

# Correlating Account on Ethereum Mixing Service via Domain-Invariant feature learning

Zheng Che, Taoyu Li, Meng Shen, *Member, IEEE*, Hanbiao Du, Liehuang Zhu, *Senior Member, IEEE*

**Abstract**—The untraceability of transactions facilitated by Ethereum mixing services like Tornado Cash poses significant challenges to blockchain security and financial regulation. Existing methods for correlating mixing accounts suffer from limited labeled data and vulnerability to noisy annotations, which restrict their practical applicability. In this paper, we propose StealthLink, a novel framework that addresses these limitations through cross-task domain-invariant feature learning. Our key innovation lies in transferring knowledge from the well-studied domain of blockchain anomaly detection to the data-scarce task of mixing transaction tracing. Specifically, we design a MixFusion module that constructs and encodes mixing subgraphs to capture local transactional patterns, while introducing a knowledge transfer mechanism that aligns discriminative features across domains through adversarial discrepancy minimization. This dual approach enables robust feature learning under label scarcity and distribution shifts. Extensive experiments on real-world mixing transaction datasets demonstrate that StealthLink achieves state-of-the-art performance, with 96.98% F1-score in 10-shot learning scenarios. Notably, our framework shows superior generalization capability in imbalanced data conditions than conventional supervised methods. This work establishes the first systematic approach for cross-domain knowledge transfer in blockchain forensics, providing a practical solution for combating privacy-enhanced financial crimes in decentralized ecosystems.

**Index Terms**—Cryptocurrency, Ethereum, mixing services, GNN.

## I. INTRODUCTION

IN recent years, Web3.0 has garnered considerable attention as a transformative paradigm for the internet [1]. With its decentralized and user-centric characteristics, Web3.0 enables users to independently manage their identity information, create digital works, and engage in digital asset transactions, thereby significantly facilitating the circulation of data value. Ethereum[2], as a vital blockchain platform underpinning the value circulation within the Web3.0 ecosystem, has also garnered significant attention from various stakeholders. According to statistical data from CoinMarketCap<sup>1</sup>, the market capitalization of Ethereum’s native token ETH exceeded \$220 billion as of April 2025.

As the most prominent mixing service provider in Ethereum, Tornado Cash (TC) [3] are designed to enhance user privacy

Z. Che is with the School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: chezheng@bit.edu.cn).

T. Li is with the School of Computer Science and Technology, Taiyuan University of Technology, JinZhong 030600, China (e-mail: 921034826@qq.com).

M. Shen, H. Du and L. Zhu are with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: {shenmeng, duhanbiao, liehuangz}@bit.edu.cn).

by obfuscating the traceability of transactions, thus making it difficult to link specific coins with their previous owners. However, this technique has raised substantial concerns regarding illicit activities, including money laundering, as well as the potential for cryptocurrencies to be misused. In February 2024, the co-founder of the video game Axy Infinity was stolen with hackers stealing 3,248 ETH and sending them to Tornado Cash to evade tracking [4]. In addition, Tornado Cash is accused of allegedly facilitating money laundering transactions amounting to almost \$1 billion on behalf of the criminal organization known as the Lazarus Group [5]. The untraceability afforded by Tornado Cash presents a significant menace to the blockchain ecosystem and financial stability. Consequently, there is an urgent imperative to dismantle the anonymity of Tornado Cash by establishing associations among the addresses involved in mixing transactions.

Figure 1 illustrates the factors contributing to the misuse of TC for criminal activities. Criminals exploit TC by submitting self-generated zk-snark promises, allowing them to transfer illicit funds into the platform’s pool. Subsequently, these funds are withdrawn into new accounts. Due to the substantial size of the TC withdrawal user base, the transfer of illegal funds to new accounts occurs discreetly, making it challenging for regulators to detect such illicit activities.

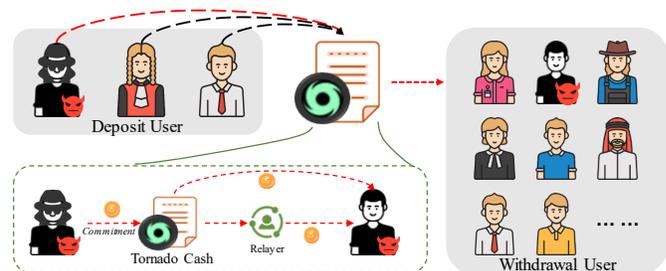


Fig. 1: Tornado Cash is abused for illegal behavior.

Recent studies have put forward various methods to identify mixing service addresses using traditional approaches, including heuristic rules [6, 7] and empirical analysis [8, 9]. However, these methods primarily rely on transaction relationships between user addresses to infer fund flow, without fully exploiting the intricate associations within address neighborhoods. Additionally, some methods based on machine learning, including deep learning [10, 11] or graph neural networks [12, 13] to detect mixing addresses. Nevertheless, these analytical techniques often necessitate extensive real-world datasets, and in the case of Tornado Cash, labeled mixing addresses are exceptionally scarce.

<sup>1</sup><https://coinmarketcap.com/>

To address the challenges of transaction traceability caused by the abuse of mixing technologies, researchers have proposed various technical approaches for tracing mixed transactions, including identifying addresses controlled by mixing services [8, 9] and correlating input-output addresses in mixed transactions [6, 7]. However, these methods heavily rely on the construction of ground-truth datasets for mixed transactions, where the quality of such datasets directly determines the accuracy of traceability models. Furthermore, the limited scale of ground-truth datasets restricts supervised learning-based traceability methods [10, 11], as insufficient labeled data hinders the establishment of comprehensive decision boundaries, leading to overfitting tendencies in complex mixing scenarios and degraded generalization performance in real-world large-scale transaction environments.

This paper proposes StealthLink, a mixing transaction traceability framework that achieves high-precision tracing under limited labeled samples while enhancing model robustness by learning cross-task invariant features between abnormal blockchain transactions and mixed transactions. Specifically, to improve association discrimination in data-scarce scenarios, we design a MixFusion module (Mixing Subgraph Fusion Encoding). This module captures local transactional behaviors and structural patterns of mixing addresses by constructing mixing subgraphs and fuses their embedded representations to generate joint features for association discrimination, thereby transforming complex transaction correlation tasks into graph classification problems. To ensure robustness against noisy data, we introduce a knowledge transfer module that enables the model to learn consistent discriminative features across the domains of abnormal transactions and mixed transactions. This alignment of cross-domain representations enhances adaptability to varying data distributions and maintains stable performance in the presence of label noise.

The primary contributions of this chapter are summarized as follows:

- First systematic analysis of the feasibility of transferring knowledge from blockchain anomaly detection to mixing transaction traceability, revealing the potential advantages of cross-domain invariant features in addressing mixing-related anonymity challenges.
- A novel framework, StealthLink, which leverages cross-task invariant feature learning to train highly robust and accurate traceability models using minimal labeled data.
- Extensive experiments on mixing transaction datasets, including few-shot learning, robustness testing, and imbalanced data scenarios. Results demonstrate that StealthLink achieves state-of-the-art performance across all evaluated conditions.

The rest of this paper is organized as follows. We first introduce the background of Tornado Cash and summarize the related work in Section II. Then, we describe the design goals in Section III. We introduce the transaction representation in Section IV and present StealthLink in Section V. Next, We evaluate the performance of StealthLink and compare it comprehensively with the state-of-the-art methods in Section VI. We conclude this paper in Section VII.

## II. BACKGROUND AND RELATED WORK

In this section, we first introduce the background of Tornado Cash, and then we summarize the recent achievements in undermining the anonymity of mixing transactions.

### A. Web3.0 and Ethereum

Web3.0 is a new generation of Internet with the concepts of de-trust, de-intermediation and digital assetisation, with blockchain as the underlying key technology, and digital production and digital consumption as the main economic forms. Web3.0 aims at data sovereign control and value circulation, and through the distributed consensus mechanism, it can completely record the process of value transfer and realise the peer-to-peer transmission of value without the need for specific intermediaries. Through smart contracts, it can form more standard and concise Distributed applications(DApp) to replace the existing Internet application services.

As the largest blockchain platform supporting smart contracts, Ethereum provides a decentralised infrastructure for building Web 3.0 DApps. To guarantee the operation of the decentralised ecosystem, Ethereum sets ETH as the fuel for smart contracts to run, encouraging miners in the network to package and maintain Ethereum transactions. As a result, ETH, as a transaction token in Web3.0, continues to receive attention from both academia and industry.

In the Ethereum blockchain, there are two types of accounts due to the presence of smart contracts: Externally Owned Accounts (EOA) and Contract Accounts (CA). EOAs function similarly to traditional bank accounts in conventional transaction systems, serving as records for transactions between users and their corresponding account balances. In contrast, CAs are internal accounts utilized by users to participate in or invoke various smart contracts. They are responsible for storing information related to smart contracts, such as bytecode and other relevant data. When transactions between EOAs involve smart contracts, they trigger transactions between CAs. Transactions initiated by EOAs are commonly referred to as external transactions, while transactions initiated by Contract Accounts are known as internal transactions.

### B. Tornado Cash

Tornado Cash, a zk-SNARK-based protocol, operates as a decentralized non-custodial mixing service. Its objective is to enhance transaction privacy by severing the link between source and destination accounts. Through the utilization of smart contracts, Tornado Cash enables the deposit of ETH and other ERC20 tokens from one account and withdrawal from another, seemingly unrelated account.

Tornado cash Proxy contract acts as a gateway for users to access the TC, and user's deposit and withdrawal actions are done by interacting with it.

**Deposit.** Before initiating a deposit, users are required to generate a private *secret* and a publicly available string *nullifier* locally. These values are then used to compute the commitment  $C$  through a hash function, such that  $C = Hash(secret|nullifier)$ . Upon initiating a deposit request

TABLE I: The comparison with the existing mixing detection methods.

Categories	Refs.	Methods	Data Source	Classifier	Method Characteristics		
					Efficient model training	Robust to noise TX.	Domain portability
Service Address Identifying	MixedSignals [8]	Empirical Study	Mixing Network Traffic	Expert Judgment	×	✓	×
	Wu et al. [9]	Heuristic Analysis	Create Mixing Transaction	Expert Judgment	×	✓	×
	Wu et al. [10]	PU Learning	Web-sourced Labels	LR	✓	×	×
	Xu et al. [13]	Ensemble Learning	Web-sourced Labels	Random Forest	×	×	×
	Moser et al. [14]	Reverse Engineering	Create Mixing Transaction	Expert Judgment	×	✓	×
	STMD [11]	Graph representation Learning	Web-Sourced Labels	MLP	×	×	×
Input-Output Correlation	Beres et al. [6]	Heuristic Analysis	None	Rule-based Judgment	×	✓	×
	Wang et al. [7]	Heuristic Analysis	Side-channel Leakage	Rule-based Judgment	×	×	×
	MixBroker [12]	Graph representation Learning	Side-channel Leakage	MLP	×	×	×
	<b>StealthLink</b>	Graph Transfer Learning	BTC malicious Dataset	MLP	✓	✓	✓

to the TC contract, users must provide the contract with the commitment  $C$  and the specified amount of funds to be deposited, denoted as  $N$ . The TC contract verifies the availability of the requested ETH amount and subsequently inserts the  $C$  as a leaf into the merkel tree list.

**Withdrawal.** Before initiating a withdrawal, the user, acting as the prover, is required to provide the TC contract, acting as the verifier, with several pieces of information to establish their ownership of the deposited funds. This includes a SNARK proof, the hash value of *nullifier*, the withdrawal account  $A$ , and the transaction fee  $f$ . The SNARK proof serves the purpose of demonstrating that the user possesses knowledge of both the merkle path of the  $C$  and the preimage of this leaf. The hash value of *nullifier* is to track spent notes, ensuring that it cannot be reused. Once the TC contract has successfully verified these information, it release the funds to  $A$ .

In the scenario where the withdrawal account  $A$  is a new account without any balance, the TC contract facilitates the transfer of funds to  $A$  through a Relayer. The Relayer, acting as an intermediary, deducts a portion of the funds as a transaction fee before transferring the remaining amount to the new account  $A$ . This transaction fee serves as compensation for the Relayer’s services in facilitating the fund transfer.

### C. Summary of Existing Studies

Mixing service have attracted increasing research attention in recent years. In this section, we briefly review the existing Mixing detection into two categories as shown in Table I.

**Service Address Identifying.** This research aims to identify transaction addresses provided by mixing services to track the flow of funds in mixing transactions. Wu et al. [9] categorized existing mixing techniques into two types, obfuscation and swapping, based on the obfuscation principle, and developed a heuristic approach to identify mixing addresses within the obfuscation mechanism. Fieke et al. [8] conducted an empirical study, centering on Bestmixer, utilizing traffic data from mixing servers and publicly available datasets on IP geographic distribution to uncover the underlying principles. Subsequently, researchers began training mixing address classifiers using machine learning models, such as Positive and Unlabeled Learning (PU learning) [10], Ensemble Learning [13], and graph representation learning [11]. However, the

reliance of machine learning models on expert experience in manual feature design poses challenges in their application to novel mixing mechanisms.

**Input-Output Correlation.** This research category aims to establish correlations among mixing accounts controlled by the same user through the observation of the user’s trading patterns within mixing activities. Two primary technical approaches are employed: heuristic rule design based on expert knowledge [6, 7], and the construction of a graph neural network-based mixing transaction prediction model, utilizing the topological features of mixing interaction graph [12]. However, the lack of ground truth sets for mixing transactions presents difficulties in validating the accuracy of the heuristics, while also limiting the precision of the correlation model.

**Summary.** There are two limitations in the existing methods. On the one hand, existing methods primarily depend on expert-rule design or supervised machine learning techniques [7–9], which generally require extensive labeled data, with the scarcity of labeled data constraining their effectiveness. On the other hand, the limited available ground datasets are built using heuristic rules, which may incorrectly associate two unrelated addresses due to unintentional actions by users (for example, the association rule based on private transactions [6] might erroneously link the sender and recipient of an airdrop transaction), thus introducing noise in the transaction data and further limiting the effectiveness of existing methods.

## III. PROBLEM DEFINITION

In this paper, we propose a cross-task transfer learning-based model for coin mixing transaction tracing. Specifically, we transfer the knowledge from malicious account detection (source domain  $\mathcal{S}$ ) to the analysis of graph-structured coin mixing transactions (target domain  $\mathcal{T}$ ), enabling effective relational inference under small-sample and noisy conditions in the target domain.

**Source Domain.** Let  $\mathcal{S} = (\mathcal{X}_S, \mathcal{Y}_S, P_S)$  denote the malicious account detection task. Each sample  $\mathbf{u}_i \in \mathbb{R}^{d_S}$  in the feature space  $\mathcal{X}_S \subseteq \mathbb{R}^{d_S}$  represents a  $d_S$ -dimensional account feature vector. The label space is defined as  $\mathcal{Y}_S = \{0, 1\}$ , where  $y_i = 1$  indicates a malicious account. The source domain contains a large-scale labeled dataset  $\mathcal{D}_S^L = \{(\mathbf{u}_i, y_i)\}_{i=1}^m$ , where all samples are drawn from the distribution  $P_S$ , and  $m$  denotes the total number of samples in the source domain.

**Target Domain.** Let  $\mathcal{T} = (\mathcal{G}_{T1}, \mathcal{G}_{T2}, \mathcal{Y}_T, P_T)$  denote the target domain, which consists of two heterogeneous coin mixing transaction graphs  $\mathcal{G}_{T1}$  and  $\mathcal{G}_{T2}$ . Specifically,  $\mathcal{G}_{T1} = (V_{\alpha1}, V_{\beta}, E_1, X_1)$  represents a transaction subgraph centered around the coin mixing account set  $V_{\alpha1} = \{v_{m1}, \dots, v_p\}$ , where  $V_{\beta} = \{v_{n1}, \dots, v_q\}$  denotes a set of normal account nodes satisfying  $V_{\alpha1} \cap V_{\beta} = \emptyset$ . The edge set  $E_1 \subseteq V_{\alpha1} \times V_{\beta}$  captures the transactional relationships between mixing and normal accounts. The node feature matrix  $X_1 \in \mathbb{R}^{p \times d_T}$  encodes  $d_T$ -dimensional features for mixing accounts.

Similarly,  $\mathcal{G}_{T2} = (V_{\alpha2}, V_{\beta}, E_2, X_2)$  denotes a separately constructed transaction subgraph, where the core node set  $V_{\alpha2}$  satisfies  $V_{\alpha2} \cap (V_{\alpha1} \cup V_{\beta}) = \emptyset$ . The label space  $\mathcal{Y}_T = \{0, 1\}$  is defined over node pairs across the two graphs: for any  $u \in V_{\alpha1} \cup V_{\alpha2}$  and  $v \in V_{\alpha1} \cup V_{\alpha2}$ ,  $y_{uv} = 1$  indicates that  $u$  and  $v$  are controlled by the same user. The labeled dataset is given by  $\mathcal{D}_T^L = \{(u_k, v_k), y_{uv}\}_{k=1}^n$ , where  $n$  denotes the total number of labeled node pairs in the target domain, and  $m \ll n$ .

The target domain is defined as  $\mathcal{T} = \{\mathcal{X}_T, \mathcal{Y}_T\}$ , where  $\mathcal{X}_T$  denotes the feature space of address pairs involved in coin mixing transactions. Each pair  $(\mathbf{v}_i, \mathbf{v}_j) \in \mathbb{R}^{d_T} \times \mathbb{R}^{d_T}$  corresponds to the feature representations of two addresses  $\mathbf{v}_i$  and  $\mathbf{v}_j$ .

The label space  $\mathcal{Y}_T = \{0, 1\}$  is a binary set indicating whether the two addresses are associated ( $y = 1$ ) or not ( $y = 0$ ).

The coin mixing traceability task is formulated as a binary classification problem. Each address pair  $(\mathbf{v}_i, \mathbf{v}_j)$  is transformed into a single feature vector  $\mathbf{x}_{ij} = \phi(\mathbf{v}_i, \mathbf{v}_j)$ , where  $\phi: \mathbb{R}^{d_T} \times \mathbb{R}^{d_T} \rightarrow \mathbb{R}^{d_T}$  is a feature fusion function, such as vector concatenation. This transformation produces a dataset  $\mathcal{D}_T = \{(\mathbf{x}_{ij}, y_{ij})\}$ , where  $\mathbf{x}_{ij} \in \mathbb{R}^{d_T}$  is the fused feature vector and  $y_{ij} \in \mathcal{Y}_T$  is the corresponding label.

However, due to the high cost of obtaining labeled data, the target domain only contains a limited number of labeled samples, denoted as  $\mathcal{D}_T^L \subset \mathcal{D}_T$ , with  $|\mathcal{D}_T^L| = n$ .

**Design Objective.** The goal is to learn a target domain mapping function  $f_T: \mathcal{X}_T \rightarrow \mathcal{Y}_T$ , where  $\mathcal{X}_T = \{\phi(u, v) \mid u, v \in V_{\alpha1} \cup V_{\alpha2}\}$  denotes the feature space of node pairs. The function  $f_T$  should satisfy the following criteria:

- **Few-shot generalization:** When the number of labeled samples is  $|\mathcal{D}_T^L| = n \leq 10$ , the model should achieve  $F_1 \geq 0.80$  on the test set.
- **Noise robustness:** Under label noise with noise rate  $\eta \leq 50\%$ , the performance degradation should be bounded by  $\frac{\Delta_{\text{clean}} - \Delta_{\eta}}{\Delta_{\text{clean}}} \leq 25\%$ .

#### IV. MOTIVATION

Malicious account detection and coin-mixing transaction tracing in blockchain-based cryptocurrencies exhibit significant overlap in their underlying knowledge domains, which provides a solid theoretical foundation for the use of cross-task transfer learning in this chapter. To validate the feasibility of transferring knowledge from the domain of malicious account detection to the domain of coin-mixing transaction association, this section presents an in-depth qualitative and quantitative analysis from the perspectives of domain knowledge and data distribution.

#### A. Qualitative Analysis of Domain Transferability

To qualitatively evaluate the feasibility of transferring knowledge from malicious behavior detection to coin mixing transaction tracing, we conduct a systematic review of recent literature in both domains [8–13, 16–21], as summarized in Table II. Our analysis focuses on four key dimensions: analytical methods, task modeling, target objects, and data characteristics.

**Analytical Methods.** Malicious behavior detection primarily relies on two types of approaches. The first involves expert-driven empirical analysis, such as taint analysis [15], which traces illicit fund flows to identify suspicious addresses. The second leverages machine learning to automate detection based on features such as transaction statistics and graph structure, using models like MLP [17] and XGBoost [20]. These methodologies are also widely adopted in coin mixing analysis, suggesting a strong methodological overlap between the two domains.

**Task Modeling.** Malicious transaction detection is generally formulated as a binary classification task (benign vs. malicious), aiming to distinguish abnormal behavioral patterns from normal transactions. Given the high anonymity of coin mixing, which is often linked to illicit activities such as money laundering, its behavioral patterns (e.g., high-frequency transactions over short periods [22]) bear strong resemblance to other forms of financial fraud. By modeling transactional similarities between sending and receiving addresses and fusing their features, coin mixing tracing can also be effectively framed as a binary classification task (associated vs. unassociated). This reveals a high degree of alignment in task modeling between the two domains.

**Target Objects.** Existing studies indicate that coin mixing technologies have become critical enablers in the ecosystem of illicit blockchain activities. Anti-money laundering investigations have consistently linked them to darknet markets, ransomware, and other illegal financial flows [4, 5]. Both tasks focus on blockchain addresses and transaction patterns as core analytical units, leveraging behavioral correlations and fund flow topologies—showing clear structural homogeneity in their analytical targets.

**Data Characteristics.** Due to the intrinsic behavioral similarities between malicious accounts and coin mixing activities, both domains exhibit strong consistency in feature engineering, whether through expert heuristics or automated learning. Key features include transaction timestamps, volumes, frequencies, gas fees, and neighborhood interactions. The high degree of overlap across these multi-dimensional feature spaces underscores the theoretical plausibility of cross-domain knowledge transfer.

#### B. Quantitative Analysis of Domain Transferability

Based on the qualitative assessment in the previous section, we further perform a quantitative analysis of the transferability from the domain of malicious transaction detection to coin mixing transaction tracing from the perspective of data distribution.

TABLE II: Task Similarity Analysis between Malicious Transaction Detection and Coin-Mixing Traceback

Domain	Method	Approach	Task Modeling	Target	Feature Attributes				
					Amount	Time	Frequency	Neighborhood	Fee
Malicious Transaction Detection	XBlockFlow [15]	Empirical Analysis	Binary Classification	Money Laundering	✓	✓	✓	✓	✗
	Xiang et al. [16]	Machine Learning	Multi-class Classification	Malicious Accounts	✓	✓	✓	✓	✗
	Bert4ETH [17]	Representation Learning	Binary Classification	Phishing Accounts	✓	✓	✓	✓	✗
	ABGRL [18]	Ensemble Learning	Binary Classification	Phishing Accounts	✓	✓	✓	✓	✓
	Chen et al. [19]	Representation Learning	Binary Classification	Ponzi Schemes	✓	✓	✓	✓	✓
	Jin et al. [20]	PU Learning	Binary Classification	Arbitrage Accounts	✓	✓	✓	✓	✗
TTAGN [21]	Representation Learning	Binary Classification	Phishing Accounts	✓	✓	✓	✓	✗	
Coin-Mixing Traceback	Wu et al. [9]	Empirical Analysis	Multi-class Classification	Mixing Transactions	✓	✓	✓	✓	✗
	Wang et al. [7]	Expert Knowledge	Binary Classification	Mixing Transactions	✓	✓	✓	✓	✓
	Wu et al. [10]	PU Learning	Binary Classification	Mixing Accounts	✓	✓	✓	✓	✗
	Xu et al. [13]	Ensemble Learning	Binary Classification	Mixing Accounts	✓	✓	✓	✓	✓
	STMD [11]	Representation Learning	Binary Classification	Mixing Accounts	✓	✓	✓	✓	✓
	MixBroker [12]	Representation Learning	Binary Classification	Mixing Transactions	✓	✓	✓	✓	✓
<b>Task Similarity</b>	<b>All tasks adopt expert-supervised learning methods with similar classification modeling, comparable targets, and highly overlapping feature sets.</b>								

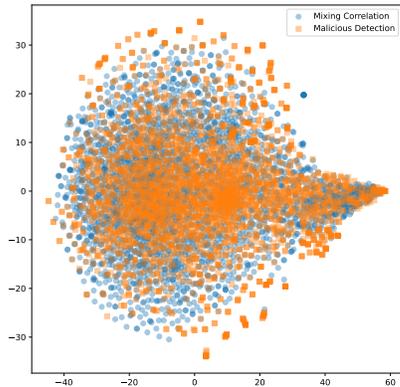


Fig. 2: The data distribution of the two domains is highly overlapping.

We adopt the Maximum Mean Discrepancy (MMD) [23] metric to evaluate the transferability between the two domains. MMD is a statistical measure used to quantify the difference between two probability distributions. Its core idea is to measure the distance between distributions by comparing the means in a Reproducing Kernel Hilbert Space (RKHS). Notably, a smaller MMD value indicates higher similarity between the two distributions.

For the coin mixing transaction tracing domain, we select the GTD dataset [12] as a representative. This dataset includes 103 pairs of associated input and output addresses from Tornado Cash transactions, each pair described by 46 features including transaction time, amount, gas price, etc. Unassociated samples are constructed by randomly shuffling these address pairs, resulting in 206 address pairs in total.

To avoid bias in MMD computation due to dataset size differences, we choose a reduced version of the BABD-13 dataset [16], denoted as BABD<sub>s</sub>, to represent the malicious behavior detection domain. The BABD<sub>s</sub> dataset is downsized by a factor of 10, containing 54,446 Bitcoin addresses covering 13 behavioral categories, with each address described by 148 features. To ensure consistency in feature dimensionality across datasets, we apply Principal Component Analysis (PCA) to reduce the features of BABD<sub>s</sub> to 46 dimensions.

Let  $P$  and  $Q$  denote the probability distributions of GTD

and BABD<sub>s</sub>, respectively. The MMD is defined as:

$$\text{MMD}(P, Q) = \|\mathbb{E}_{x \sim P}[\phi(x)] - \mathbb{E}_{y \sim Q}[\phi(y)]\|_{\mathcal{H}} \quad (1)$$

The computed MMD value between the two domains is  $2.37 \times 10^{-5}$ , which is remarkably small. In practice, MMD values exceeding 0.01 typically indicate significant distributional differences [23]. Hence, this result suggests that the two domains exhibit highly similar feature distributions. We further visualize the data distributions from both domains using the t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm. As shown in Figure 2, the data from the two domains show substantial overlap, which corroborates the low MMD value and provides strong quantitative evidence for the similarity in feature distributions between the two tasks.

## V. STEALTHLINK

In this section, we present the detailed design of *StealthLink*, a method that leverages cross-task knowledge transfer to achieve high-precision tracing of coin mixing transactions under limited labeled samples. The system architecture of StealthLink is illustrated in Figure 3.

### A. Overview of the Proposed Approach

StealthLink consists of three main components: the Mixed Transaction Subgraph Fusion Embedding Module, the Cross-Task Knowledge Transfer Module, and the Mixed Account Association Discrimination Module.

**Mixed Transaction Subgraph Fusion Embedding.** This module leverages the local topological structure of mixing transaction addresses and graph embedding techniques to capture the latent interaction patterns among addresses involved in coin mixing. It constructs fused sample representations that express the associations between mixed addresses, thereby transforming the complex tracing task into a graph classification problem.

**Cross-Task Knowledge Transfer.** This module introduces a cross-task feature decoupling mechanism to guide the model in learning task-invariant and discriminative shared feature representations. It implicitly aligns the feature space between coin mixing samples and malicious account detection samples, enabling effective cross-task knowledge transfer.

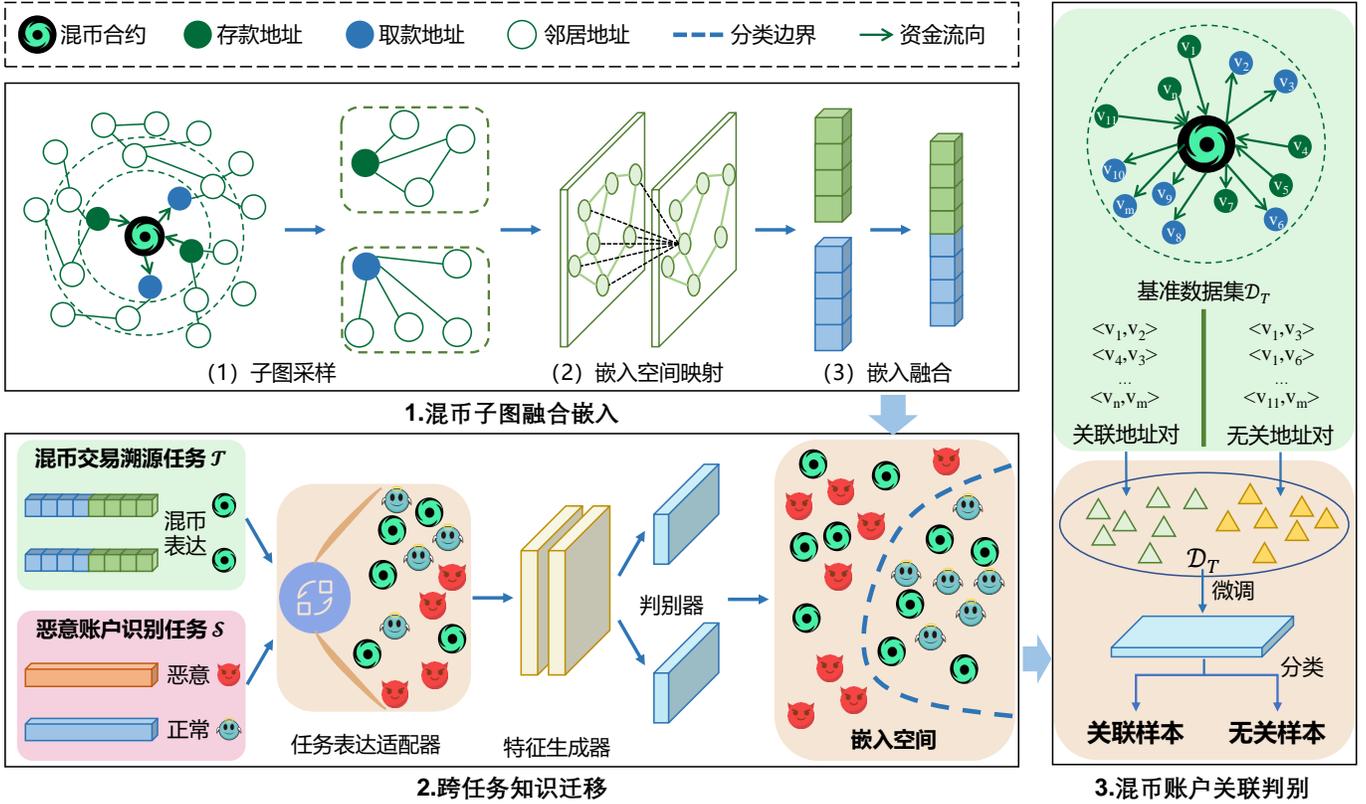


Fig. 3: Overview of the StealthLink System

**Mixed Account Association Discrimination.** Based on the pre-trained encoder obtained from the knowledge transfer module, this component employs a few-shot learning strategy to supervise the training of the discriminator. By integrating both transferred and task-specific features, it builds a robust model for tracing coin mixing transactions.

### B. Mixed Transaction Subgraph Fusion Embedding

Existing studies on coin mixing transaction association typically analyze the transaction features of individual addresses independently. This approach faces two main limitations. First, the association between mixed addresses is often implicitly embedded in the local behavioral patterns and subgraph structures of both addresses, involving complex nonlinear interactions. Analyzing the transaction attributes of a single address in isolation often fails to capture the combinatorial patterns between address features [6]. Second, due to transaction sparsity (e.g., newly created addresses) and the obfuscation characteristics of mixing techniques (e.g., generating multiple outputs of equal value) [12], independent address feature analysis often lacks reliable information to determine transaction associations, thereby limiting the effectiveness of standalone embedding methods.

To address these issues, we propose a fusion embedding method tailored for coin mixing subgraphs. First, we use a  $k$ -hop neighborhood sampling strategy to construct local subgraphs for each of the two addresses involved in a coin mixing transaction, capturing structural and behavioral features in

their respective neighborhoods. Second, we employ a graph neural network (GNN) to encode each subgraph, enabling efficient extraction of local representations. Finally, we fuse the two address embeddings through concatenation to generate a joint representation that reflects bidirectional interaction patterns. By integrating the subgraph representations of mixed addresses, the coin mixing association task is formulated as a graph classification problem—determining whether a given joint representation corresponds to an associated address pair.

Specifically, given a coin mixing address pair  $v_\alpha \in V_{\alpha 1}$  and  $v_\beta \in V_{\alpha 2}$ , we apply a  $k$ -hop neighborhood sampling strategy to construct their respective local subgraphs from the blockchain transaction network:  $\mathcal{G}_\alpha = (V_\alpha, E_\alpha, X_\alpha)$  and  $\mathcal{G}_\beta = (V_\beta, E_\beta, X_\beta)$ , where  $V_\alpha = \{v_\alpha\} \cup \mathcal{N}_k(v_\alpha)$  contains the mixed address  $v_\alpha$  and its  $k$ -hop neighbors, and  $X_\alpha \in \mathbb{R}^{|V_\alpha| \times d_T}$  is the node feature matrix. Each subgraph is then encoded using a GNN. The embedding of a node  $v_i$  at the  $l$ -th layer is computed as:

$$h_i^{(l+1)} = \sigma \left( W^{(l)} \cdot \text{AGG} \left( \{h_i^{(l)}\} \cup \{h_j^{(l)} : j \in \mathcal{N}(i)\} \right) \right) \quad (2)$$

where  $W^{(l)}$  is the weight matrix at layer  $l$ ,  $\sigma(\cdot)$  is a nonlinear activation function, and  $\text{AGG}(\cdot)$  denotes the aggregation function over the neighborhood. After  $L$  layers of message passing, we obtain the high-level embeddings  $h_\alpha$  and  $h_\beta$  for subgraphs  $\mathcal{G}_\alpha$  and  $\mathcal{G}_\beta$ , respectively. Finally, the two embeddings are concatenated to form a joint representation:

$$\tilde{h} = [h_A | h_B] \in \mathbb{R}^{2d_C} \quad (3)$$

where  $d_C$  is the embedding dimension for each subgraph. This joint representation comprehensively integrates transaction-level and local structural features of both addresses, enabling subsequent modules to learn domain-invariant feature representations.

### C. Cross-task Knowledge Transfer

After constructing the joint representation samples for coin-mixing transactions, this section introduces a feature disentanglement mechanism across tasks to guide the model in learning discriminative and shared representations, thereby achieving implicit alignment between coin-mixing transaction samples and malicious account detection samples in the feature space. This enables knowledge transfer across tasks.

Due to significant differences in data representation and feature dimensions between the coin-mixing joint representations and the malicious account detection task—as well as differing focuses on blockchain transaction characteristics and behavior patterns—it is insufficient to directly reuse the representations learned from the malicious account detection task for coin-mixing tracing. To address this, we design a cross-task knowledge transfer module, which consists of two main components: (1) a task representation adapter that maps malicious account detection samples to a feature space compatible with the coin-mixing tracing task; and (2) cross-task invariance learning, which generates domain-aligned and task-discriminative representations in the adapted feature space. The detailed process is as follows:

First, we compute the mean feature vector  $\mu_S$  of all encoded samples from the malicious account detection task using encoder  $E(\cdot)$  as shown in Eq. 4:

$$\mu_S = \mathbb{E}_{\mathbf{u} \sim P_S} [E(\mathbf{u})] \quad (4)$$

where  $E(\cdot) : \mathbb{R}^{d_S} \rightarrow \mathbb{R}^{d_S}$  denotes the encoder,  $\mathbf{u} \in \mathcal{X}_S$  is a sample from the source domain, and  $\mathbb{E}_{\mathbf{u} \sim P_S}$  represents the expectation over the source distribution  $P_S$ . The encoder output  $[E(\mathbf{u})] \in \mathbb{R}^{d_S}$  is a high-dimensional feature representation.

Next, a task representation adapter  $\mathcal{T}(\cdot)$  is introduced to project the feature vectors from the malicious account detection task into a lower-dimensional space compatible with the coin-mixing tracing task. The transformation is defined in Eq. 5:

$$\mathcal{T}(\mathbf{u}_i) = U^\top \left( E(\mathbf{u}_i) - \mu_S \right) \quad (5)$$

Here,  $U \in \mathbb{R}^{d_S \times d_P}$  is a trainable projection matrix with  $d_P = 2d_C$ , and  $\mathcal{T}(\cdot) : \mathbb{R}^{d_S} \rightarrow \mathbb{R}^{d_P}$  performs the cross-task feature alignment, ensuring that the adapted feature vectors match the dimensionality of the joint representation  $\tilde{h} \in \mathbb{R}^{2d_C}$  from the coin-mixing tracing task.

After obtaining the task-aligned representations, we apply a discrepancy-based transfer learning approach to capture invariant features across tasks. This transfer learning process involves training a feature generator  $F(\cdot) : \mathbb{R}^{d_P} \rightarrow \mathbb{R}^{d_C}$  along with two discriminators  $C_1(\cdot)$  and  $C_2(\cdot)$ . Initially, the feature generator  $F(\cdot)$  is fixed, and the discriminators are trained to maximize the prediction discrepancy on samples from the coin-mixing tracing task. This encourages the discriminators

to make diverse predictions in the current feature space. The discrepancy loss is defined as follows:

$$\mathcal{L}_{\text{dis}} = \mathbb{E}_{\tilde{h} \sim P_T} [\|C_1(\tilde{h}) - C_2(\tilde{h})\|_1] \quad (6)$$

where  $P_T$  denotes the target distribution (coin-mixing tracing task), and  $\|\cdot\|_1$  is the L1 norm.

Next, we fix the discriminators and train the feature generator  $F(\cdot)$  to minimize the discrepancy while also maintaining classification performance on the source domain. The optimization objective is given by Eq. 7:

$$\begin{aligned} \mathcal{L}_{\text{gen}} = & \mathbb{E}_{\tilde{h} \sim P_T} [\|C_1(\tilde{h}) - C_2(\tilde{h})\|_1] \\ & + \lambda \cdot \mathbb{E}_{(\mathbf{u}, y) \sim \mathcal{D}_S^L} [\mathcal{L}_{\text{ce}}(C_1(F(\mathcal{T}(\mathbf{u}))), y)] \end{aligned} \quad (7)$$

where  $\mathcal{L}_{\text{ce}}(p, y)$  denotes the cross-entropy loss,  $\lambda > 0$  is a trade-off parameter balancing the classification loss and the domain alignment loss, and  $\mathcal{D}_S^L = \{(\mathbf{u}, y)\}$  is the labeled dataset from the source domain.

Through this learning process of task-invariant features, the feature generator  $F(\cdot)$  is able to produce representations that are both highly discriminative and domain-aligned, enabling effective knowledge transfer from the malicious account detection task to the coin-mixing tracing task. This ultimately enhances the performance of the coin-mixing tracing model in scenarios with limited labeled data.

### D. Mixer Account Association Classification

The mixer account association classification module aims to construct a final discriminator for mixer transaction tracing, based on the task-aligned feature generator trained in the previous module. Building upon the aforementioned task expression adapter, this module introduces a mixer account association classifier  $C(\cdot)$ , which is fine-tuned using a small labeled set of mixer transaction tracing data  $\mathcal{D}_T^L = \{(\tilde{h}_j, y_j)\}_{j=1}^n$ . During fine-tuning, to ensure the stability of feature representations, the parameters of the feature generator are kept frozen, and only the parameters of the classifier are updated.

For each pair of mixer transaction accounts  $(u_j, v_j)$ , a fused feature vector  $\tilde{h}_j \in \mathbb{R}^{2d_C}$  is obtained via the task expression adapter module, and then passed through the classifier  $C(\cdot)$  to produce the association prediction. The objective during fine-tuning is to minimize the supervised classification loss, which is computed as shown in Equation 8:

$$\mathcal{L}_{\text{ce}}^{(T)} = -\frac{1}{n} \sum_{j=1}^n \left[ y_j \log C(\tilde{h}_j) + (1 - y_j) \log (1 - C(\tilde{h}_j)) \right], \quad (8)$$

where  $y_j \in \{0, 1\}$  indicates whether an association exists between the account pair. Through fine-tuning, the classifier  $C(\cdot)$  can effectively leverage the learned cross-task invariant features to accurately determine the association between mixer accounts.

## VI. EXPERIMENTS

In this section, we evaluate the effectiveness of StealthLink on existing datasets using standard metrics, including accuracy, recall, and F1-score. We compare its performance with eight

state-of-the-art mixer transaction tracing methods. The experiments demonstrate that StealthLink achieves the following four key capabilities:

- (1) In few-shot learning scenarios, StealthLink achieves accurate and robust mixer transaction tracing, outperforming existing models (see Section VI-B);
- (2) In scenarios with pseudo-associated noisy samples, StealthLink maintains high accuracy and robustness in tracing, surpassing baseline models (see Section VI-C);
- (3) In imbalanced dataset scenarios, StealthLink demonstrates precise and robust tracing capabilities, outperforming current approaches (see Section VI-D);
- (4) Each component of StealthLink contributes to the overall performance in identifying associations between mixer addresses (see Section VI-E).

### A. Preliminary

**Experimental Environment.** All experiments were conducted on a Linux-based server equipped with a 16-core Xeon(R) Platinum 8352V processor, 90GB of system memory, and an NVIDIA GeForce RTX 4090 GPU (24GB VRAM) with driver version 560.35.03. The software environment includes Python 3.7 and PyTorch 1.13.1, with CUDA Toolkit 11.7.

**Datasets.** This section involves three types of datasets: the Bitcoin malicious account detection dataset, the mixer transaction dataset, and the mixer benchmark dataset.

- **Bitcoin Malicious Account Detection Dataset.** This dataset is constructed based on the BABD-13 dataset [16], which contains fine-grained labels for six types of malicious accounts (phishing, gambling, darknet markets, blacklisted addresses, money laundering, and Ponzi schemes) and seven types of benign accounts. In our experimental design, the six malicious account types are merged into a single malicious class, forming a binary classification task together with the benign accounts. The final dataset contains 41,662 benign accounts and 12,842 malicious accounts, totaling 54,504 samples.
- **Tornado Cash Transaction Dataset.** We used the Etherscan API to crawl all mixing transaction data from the launch of Tornado Cash up to March 31, 2022. This resulted in 30,823 deposit addresses and 44,814 withdrawal addresses. Any combination of a deposit and a withdrawal address is treated as an unlabeled mixer transaction sample.
- **Mixing Benchmark Dataset.** This dataset aggregates labeled Ethereum address pairs from the studies [6, 12], consisting of a total of 291 associated address pairs.

**Baseline Methods.** To comprehensively evaluate the performance of StealthLink, we compare it with eight state-of-the-art address association methods for mixer transactions. All baselines were fine-tuned to ensure optimal performance on the evaluation datasets.

- **Gas Fingerprinting (GF)** [7]: A heuristic method that matches transactions by identifying cases where the last 9 digits of the gas price are identical in both sending and receiving transactions.

- **Cross Contract Correlation (CC)** [7]: Another heuristic method that links deposit and withdrawal addresses across multiple mixer pools, leveraging the fixed denomination feature of Tornado Cash and user behaviors involving multiple deposits.
- **DeepWalk** [24]: Generates random walks in the transaction subgraph to produce sequences of nodes, which are then embedded into a low-dimensional vector space. Similarity between embeddings is used to infer address associations.
- **Node2Vec** [25]: Extends DeepWalk with two hyperparameters that balance breadth-first and depth-first sampling, achieving better trade-offs between local and global structural features.
- **GAT** [26]: Utilizes self-attention mechanisms to assign different weights to neighboring nodes in the transaction subgraph of a mixer address, generating low-dimensional embeddings. Address similarity is then assessed via these embeddings.
- **GIN** [27]: Aggregates neighborhood information in the transaction subgraph using multi-layer perceptrons (MLPs), capturing subtle subgraph structural differences through low-dimensional embeddings.
- **GraphSAGE** [28]: Samples and aggregates neighborhood node information in the transaction subgraph to produce embeddings, which are then used to evaluate address similarity.
- **MixBroker** [12]: Models Tornado Cash address relationships via interaction graphs, and inputs statistical features into a GNN-based classifier to analyze address associations.

**Parameter Settings.** For StealthLink, we adopt a two-stage training strategy. In the pretraining stage, we use a Transformer-based encoder consisting of 3 stacked layers with parameters:  $d_{\text{model}} = 92$ ,  $n_{\text{head}} = 4$ . The output features are projected via a head composed of linear layers, batch normalization (BN), and ReLU activation. The pretraining uses SGD optimizer with a learning rate of  $1 \times 10^{-4}$ , momentum of 0.9, and weight decay of 0.0005. During the fine-tuning stage, the encoder’s output embeddings are frozen and passed to a multi-layer perceptron (MLP) classifier. The MLP consists of four hidden layers with 1024 units each, an input dimension of 92, an output dimension of 2, ReLU activation for hidden layers, and a Sigmoid activation at the output layer.

For DeepWalk and Node2Vec, the embedding dimension is set to 43, with walk length, number of walks, and context window size all set to 5. Node2Vec’s return and in-out parameters are both set to 0.75, using the Skip-gram model ( $sg=1$ ) with 4 worker threads.

For GAT, the input feature dimension is set to 31, hidden layer dimension to 43, number of classes to 30, and number of attention heads to 4.

For GIN, the model consists of 2 layers, each with a 3-layer MLP. Input, hidden, and output dimensions are 31, 256, and 43 respectively. Both `graph_pooling_type` and `neighbor_pooling_type` are set to "mean", and `final_dropout` is 0.01.

For GraphSAGE, the number of input channels is 31, hidden

channels is 128, number of layers is 2, and output channels is 43.

For MixBroker, the model is configured with GNN\_NET, where the GNN comprises two SAGEConv layers: the first maps input features to a 32-dimensional hidden space, and the second maps to a 16-dimensional output. The optimizer is Adam, with a learning rate of 0.01.

### B. Few-shot Learning Evaluation

This section presents a systematic evaluation of StealthLink’s performance under few-shot learning scenarios. Specifically, we conduct a quantitative analysis of its capability in link prediction for mixing transactions when the training set consists of only 1, 3, 5, or 10 samples. To mitigate the impact of sample selection randomness, we generate 10 independent training sets for each sample size via random resampling. The final results are reported as the mean  $\pm$  standard deviation across the 10 trials.

To establish performance baselines, we also evaluate the model trained on the full labeled dataset of mixing transactions. It is important to note that rule-based methods (GF and CC), which do not involve parameter training, are evaluated only under the full-data setting. The quantitative results under different experimental configurations are summarized in Table III.

As shown in Table III, StealthLink demonstrates consistently strong discriminative performance across all few-shot scenarios. This superior performance can be attributed to the combination of cross-task invariant feature learning and the task representation adapter module, which enables the model to effectively activate large-scale knowledge of malicious account detection even with very limited mixing transaction samples, thereby significantly improving accuracy under low-resource conditions. Remarkably, even with as few as  $N = 3$  labeled samples, StealthLink achieves an F1 score of 0.9580, which significantly outperforms all other baselines trained on the full dataset. For example, MixBroker only achieves an F1 score of 0.8122 under full supervision, indicating the superior capability of the proposed method in few-shot learning scenarios for mixing transaction tracing.

In contrast, two heuristic-based methods, GF and CC, exhibit poor performance. Under full supervision, GF and CC achieve F1 scores of only 0.2226 and 0.1275, respectively. This underperformance is due to the heuristic methods relying solely on local and isolated features for address linkage in mixing transactions, making them incapable of adapting to the complex and dynamic structure of mixing transaction networks. Consequently, they struggle to extract deeper feature representations, resulting in limited scalability and effectiveness in large-scale mixing transaction tracing tasks.

### C. Robustness Evaluation under Noisy Labels

This section systematically evaluates the model’s robustness under label noise. Given that existing annotated datasets are constructed based on heuristic rules [6, 12], they inherently contain mislabeled address association pairs in mixing transactions. To address this, we construct a controlled noise injection

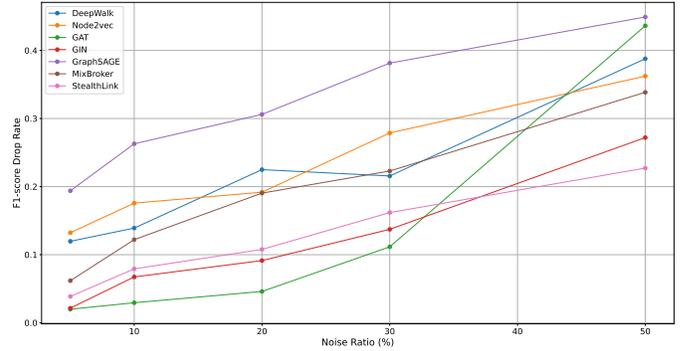


Fig. 4: Relationship between label noise rate and performance degradation rate

environment by introducing 5%–50% of false association samples into the training set. This allows us to quantitatively analyze the performance stability of StealthLink in the task of mixing address association under noisy conditions. A ten-fold cross-validation strategy is employed, and the model’s performance under noise interference is reported in the form of mean  $\pm$  standard deviation. Table IV presents the comparative results of quantitative evaluations under different levels of noise.

As shown in Table IV, StealthLink demonstrates a significant robustness advantage on datasets contaminated with pseudo-associated noisy labels. Under all noise ratios, StealthLink consistently achieves higher F1-scores compared to other methods, outperforming the second-best approach, MixBroker, by 15%–20%. This improvement is attributed to the model’s ability to learn general representations through cross-task invariant feature learning, combined with a dynamic adversarial learning framework consisting of a feature generator and dual discriminators. This architecture effectively filters out noise interference, thereby significantly enhancing the model’s discriminative capability under high-noise conditions.

To further evaluate the noise resistance of StealthLink, we analyze the relationship between label noise rates and performance degradation rates, as shown in Fig. 4. It can be observed that StealthLink exhibits a markedly lower slope in its performance degradation curve compared to baseline methods. Specifically, its F1-score degradation rate remains below 0.25, indicating stronger robustness to label noise. These results confirm the stability and robustness of StealthLink in noisy environments.

### D. Evaluation on Imbalanced Datasets

This section further evaluates the performance of StealthLink in the task of association discrimination under imbalanced positive and negative sample training scenarios. To control the influence of sample randomness, random sampling is used, and the mean is calculated based on 10-fold cross-validation. The classification results are tested by constructing four different positive-to-negative sample ratios: 1:5, 1:10, 1:15, and 1:25. The experimental results are shown in Table V.

TABLE III: Performance Comparison Under Few-Shot Scenarios

N	Metric	Method								
		GF	CC	DeepWalk	Node2vec	GAT	GIN	GraphSAGE	MixBroker	StealthLink
1	Accuracy	–	–	0.4309±0.0264	0.4193±0.0353	0.4832±0.1163	0.7155±0.0904	0.4865±0.0326	0.7851±0.1138	0.7879±0.1288
	Recall	–	–	0.3979±0.0627	0.4382±0.1689	0.4774±0.3142	0.6776±0.1404	0.6215±0.2460	0.7277±0.2409	0.8324±0.0484
	F1-score	–	–	0.4138±0.0325	0.4285±0.0689	0.4803±0.2486	0.6788±0.0421	0.5458±0.0980	0.7554±0.1453	<b>0.8096±0.0678</b>
3	Accuracy	–	–	0.4295±0.0332	0.4187±0.0447	0.7264±0.0840	0.8642±0.0751	0.5491±0.0365	0.7881±0.1179	0.9429±0.0199
	Recall	–	–	0.4062±0.0670	0.4579±0.1420	0.7079±0.0801	0.7048±0.0872	0.5140±0.1260	0.7515±0.1732	0.9738±0.0167
	F1-score	–	–	0.4175±0.0270	0.4375±0.0772	0.7170±0.0451	0.7713±0.0562	0.5310±0.0633	0.7694±0.0959	<b>0.9580±0.0118</b>
5	Accuracy	–	–	0.4591±0.0339	0.4490±0.0186	0.7021±0.0560	0.8563±0.0864	0.5830±0.0320	0.7986±0.2921	0.9657±0.0085
	Recall	–	–	0.4706±0.0986	0.4632±0.0935	0.7395±0.0885	0.7790±0.0873	0.5227±0.1263	0.8106±0.2857	0.9660±0.0104
	F1-score	–	–	0.4648±0.0375	0.4560±0.0543	0.7203±0.0217	0.8084±0.0475	0.5512±0.0647	0.8026±0.2877	<b>0.9657±0.0044</b>
10	Accuracy	–	–	0.4564±0.0300	0.4654±0.0148	0.7323±0.0448	0.8313±0.0257	0.6452±0.4437	0.8988±0.0783	0.9741±0.0031
	Recall	–	–	0.3822±0.0557	0.4266±0.0858	0.7597±0.0623	0.7552±0.0705	0.5982±0.0739	0.7409±0.1027	0.9662±0.0036
	F1-score	–	–	0.4161±0.0304	0.4452±0.0445	0.7458±0.0202	0.7916±0.0422	0.6208±0.0468	0.8122±0.1076	<b>0.9698±0.0015</b>
ALL	Accuracy	0.9291	1.0000	0.4979±0.1478	0.4939±0.1698	0.8808±0.1491	0.8512±0.0274	0.8006±0.0508	0.8781±0.1040	0.9832±0.0082
	Recall	0.1265	0.0681	0.5562±0.2549	0.6140±0.2787	0.6971±0.3785	0.7982±0.0186	0.8848±0.0484	0.8473±0.1250	0.9913±0.0057
	F1-score	0.2226	0.1275	0.5255±0.1912	0.5474±0.2117	0.7782±0.3380	0.8239±0.0130	0.8403±0.0274	0.8548±0.0740	<b>0.9872±0.0103</b>

TABLE IV: Performance Comparison under Different Noise Ratios

Noise Ratio	Metric	Method						
		DeepWalk	Node2vec	GAT	GIN	GraphSAGE	MixBroker	StealthLink
5%	Accuracy	0.4538±0.0293	0.4513±0.0224	0.8541±0.0250	0.8468±0.0370	0.6501±0.0315	0.8822±0.0462	<b>0.9665±0.0103</b>
	Recall	0.4717±0.0919	0.5014±0.1126	0.6891±0.0439	0.7693±0.0849	0.7072±0.1322	0.7351±0.1000	<b>0.9321±0.0951</b>
	F1 Score	0.4626±0.0493	0.4750±0.0582	0.7627±0.0144	0.8062±0.0437	0.6774±0.0467	0.8020±0.0615	<b>0.9490±0.0526</b>
10%	Accuracy	0.4249±0.0323	0.4594±0.0290	0.8734±0.0264	0.8343±0.0501	0.5909±0.0279	0.8516±0.0167	<b>0.9165±0.0103</b>
	Recall	0.4837±0.1656	0.4434±0.1244	0.6653±0.0439	0.7120±0.1202	0.6508±0.1575	0.6706±0.0334	<b>0.9017±0.0015</b>
	F1 Score	0.4524±0.0941	0.4512±0.0706	0.7553±0.0196	0.7684±0.0662	0.6194±0.0663	0.7504±0.0272	<b>0.9090±0.0052</b>
20%	Accuracy	0.4071±0.0316	0.4371±0.0263	0.8113±0.0331	0.8376±0.0506	0.5373±0.0384	0.7902±0.0042	<b>0.8896±0.0158</b>
	Recall	0.4074±0.1846	0.4477±0.1445	0.6842±0.0771	0.6764±0.1366	0.6375±0.2147	0.6154±0.0261	<b>0.8721±0.0137</b>
	F1 Score	0.4073±0.1095	0.4424±0.0834	0.7424±0.0225	0.7486±0.0729	0.5831±0.1079	0.6920±0.0266	<b>0.8808±0.0080</b>
30%	Accuracy	0.4263±0.0281	0.3791±0.0346	0.7686±0.0482	0.8048±0.0500	0.4628±0.0333	0.7633±0.0051	<b>0.8331±0.0137</b>
	Recall	0.3989±0.1533	0.4117±0.2087	0.6283±0.1033	0.6366±0.1407	0.5925±0.3000	0.5876±0.0083	<b>0.8217±0.0108</b>
	F1 Score	0.4122±0.0913	0.3948±0.1396	0.6914±0.0465	0.7108±0.0719	0.5198±0.2005	0.6643±0.0083	<b>0.8274±0.0128</b>
50%	Accuracy	0.3114±0.0457	0.3644±0.0300	0.4297±0.0897	0.6177±0.0386	0.4528±0.0342	0.6842±0.0290	<b>0.8001±0.1306</b>
	Recall	0.3325±0.2370	0.3351±0.1647	0.4480±0.2148	0.5826±0.2431	0.4735±0.2015	0.4818±0.0580	<b>0.7285±0.1412</b>
	F1 Score	0.3217±0.1718	0.3491±0.1325	0.4387±0.1533	0.5998±0.1614	0.4629±0.1014	0.5654±0.0533	<b>0.7629±0.0771</b>

TABLE V: Performance Comparison on Imbalanced Datasets

Positive-to-Negative Ratio	Evaluation Metric	Methods						
		GIN	GAT	GraphSAGE	DeepWalk	Node2Vec	MixBroker	StealthLink
1:5	Accuracy	0.8361±0.0024	0.8351±0.2746	0.7979±0.0193	0.3087±0.1068	0.7063±0.3529	0.8961±0.0695	0.9741±0.0031
	Recall	0.7569±0.0945	0.4761±0.2179	0.8163±0.1338	0.1913±0.0548	0.3326±0.1594	0.7460±0.0960	0.9162±0.0036
	F1 Score	0.7944±0.0567	0.6065±0.2547	0.8070±0.0844	0.2362±0.0718	0.4522±0.2190	0.8094±0.0538	<b>0.9443±0.0015</b>
1:10	Accuracy	0.7964±0.0011	0.7706±0.3554	0.7858±0.0125	0.3977±0.2364	0.7072±0.3676	0.9213±0.0818	0.9257±0.0085
	Recall	0.7139±0.0599	0.4334±0.2700	0.6903±0.2167	0.1183±0.0642	0.2677±0.1488	0.7181±0.0742	0.9260±0.0104
	F1 Score	0.7530±0.0335	0.5548±0.3293	0.7367±0.1676	0.1824±0.0988	0.3884±0.2116	0.8033±0.0530	<b>0.9258±0.0044</b>
1:15	Accuracy	0.7164±0.2988	0.7801±0.3413	0.7323±0.0131	0.4910±0.2956	0.7143±0.3649	0.8924±0.0658	0.9129±0.0199
	Recall	0.6817±0.2911	0.4368±0.2668	0.6537±0.2032	0.1069±0.0610	0.2528±0.1498	0.6942±0.0747	0.9038±0.0167
	F1 Score	0.6987±0.2947	0.5600±0.3237	0.6908±0.1709	0.1756±0.0950	0.3734±0.2135	0.7789±0.0602	<b>0.9085±0.0118</b>
1:25	Accuracy	0.6972±0.2990	0.8048±0.3325	0.7302±0.0093	0.5946±0.3688	0.7764±0.3447	0.8826±0.0489	0.7879±0.1288
	Recall	0.6637±0.2884	0.4315±0.2681	0.5250±0.1640	0.1099±0.0670	0.2352±0.1452	0.6768±0.0715	0.8324±0.0484
	F1 Score	0.6801±0.2934	0.5618±0.3272	0.6109±0.1504	0.1856±0.1129	0.3611±0.2083	0.7634±0.0484	<b>0.8096±0.0678</b>

As shown in Table V, StealthLink maintains excellent discriminatory performance even under extreme class imbalance conditions. Specifically, as the positive-to-negative sample ratio increases from 1:5 to 1:25, its F1-score exhibits a stepwise decline (0.9443  $\rightarrow$  0.8096), but it still consistently outperforms the baseline models. Meanwhile, the second-best benchmark model, MixBroker, experiences a larger F1-score drop of 9.8% (0.8094  $\rightarrow$  0.7634) under the same test conditions, resulting in a significant difference when compared to StealthLink. This phenomenon can be attributed to the synergistic effect of StealthLink’s cross-task invariance feature learning and adversarial discriminator difference minimization, which effectively preserves and aligns key discriminative features even in highly imbalanced conditions, thus balancing high precision and recall. This enables StealthLink to robustly perform coin-mixing transaction link prediction, even when minority class samples are extremely scarce.

### E. Ablation Study

This section quantitatively analyzes the impact of different components on the performance of StealthLink, including the task expression adapter, feature generator, and transfer learning paradigm in the cross-task knowledge transfer module, as well as the classifier in the coin-mixing account linkage module. Since the framework design focuses on coin-mixing entity linkage under small-sample training scenarios, this experiment primarily evaluates the impact of different components on StealthLink’s performance in such small-sample learning contexts.

**Task Expression Adapter Evaluation.** The task expression adapter aligns the feature space of malicious account detection and coin-mixing transaction tracing tasks through feature mapping strategies. This section evaluates the impact of three common feature mapping strategies on StealthLink’s performance: dimensionality expansion, PCA dimensionality reduction, and contribution pruning. Dimensionality expansion projects the low-dimensional source domain representation to the target domain’s dimension using a Multi-Layer Perceptron (MLP); PCA dimensionality reduction compresses the high-dimensional source domain features to the target dimension using Principal Component Analysis (PCA); contribution pruning iteratively removes low-importance features based on feature contribution ranking until the target dimension is reached. We evaluate the model performance under each strategy using ten-fold cross-validation, with the results presented in Table VI.

As shown in Table VI, the PCA dimensionality reduction strategy achieves the highest F1 score across all data size settings, indicating that the PCA reduction strategy can maintain the model’s coin-mixing transaction address association ability in scenarios with varying label sparsity. This is because PCA effectively extracts the core feature distribution shared by the malicious account detection task and the coin-mixing transaction tracing task by retaining the principal components with the largest global variance. It reduces noise dimensions while preserving the integrity of key features. In contrast, the dimensionality expansion strategy performs

poorly in scenarios with a small number of label samples, likely because its reliance on the MLP trained with limited labeled data leads to overfitting or difficulties in capturing the feature differences between the two tasks. Meanwhile, contribution pruning may lose some subtle but discriminative feature dimensions when pruning low-contribution features, leading to an overall performance decline.

**Feature Generator Evaluation.** The feature generator generates shared feature representations with cross-task discriminative ability for samples from two different tasks through adversarial training strategies. This section evaluates the impact of six commonly used feature generators in blockchain anomaly detection and tracing research on StealthLink’s performance: ResNet-18, ResNet-50, ResNet-101 [29], MLP, LSTM, and Transformer [30]. To ensure the reliability of the evaluation, we conducted a systematic test of the performance of all feature generators using ten-fold cross-validation, with results presented in Table VII.

**Transformer as a Feature Generator.** The Transformer demonstrates significant advantages as a feature generator when the sample size  $N \geq 3$ . Its F1-score continues to improve as the data size increases, surpassing 90% when  $N = 10$ . This result validates the effectiveness of the Transformer in capturing shared features across tasks. This is because the Transformer can calculate global node association weights through multi-head attention mechanisms, enabling it to precisely capture implicit behavioral patterns in address interactions across different tasks. At the same time, the standard deviation of the Transformer decreases as the sample size grows, indicating that its performance stability improves significantly with the sufficiency of training data.

**Transfer Learning Paradigm Evaluation.** The transfer learning paradigm ensures the effective transfer of domain knowledge from the malicious account task to the cross-task migration of mixed coin transaction tracing. In this section, we compare models without the transfer learning paradigm, denoted as "w/o Transfer," with those employing different transfer learning paradigms, including: DAN (Domain Adversarial Networks) [31], DANN (Domain-Adversarial Neural Network) [32], and MCD (Maximum Classifier Discrepancy) [33]. To ensure the reliability of the evaluation, we use ten-fold cross-validation to assess the performance of models with different feature generators. The results are shown in Table VIII.

As shown in Table VIII, the transfer learning framework significantly enhances the performance of StealthLink. Specifically, the "w/o Transfer" model performs significantly worse than the models using transfer learning frameworks in all sample settings. The F1 score is reduced by approximately 5% to 30% compared to the transfer learning models, proving that the transfer learning framework plays a crucial role in transferring domain knowledge from the malicious account task to the coin mixing transaction tracing task.

Among the transfer learning frameworks, MCD demonstrates the most significant effect on task knowledge transfer. Specifically, it achieves an F1 score of approximately 0.81 with only a small number of labeled samples ( $N=1$ ), which is about 10% higher than the F1 scores of the other two transfer

TABLE VI: Performance Comparison of StealthLink with Different Feature Mapping Strategies

N	Evaluation Metric	Alignment Method		
		Dimensionality Expansion	PCA	Contribution Pruning
1	Accuracy	0.6672±0.2501	0.7879±0.1288	0.5969±0.1733
	Recall	0.6759±0.1372	0.8324±0.0484	0.7655±0.2329
	F1 Score	0.6715±0.1004	<b>0.8096±0.0678</b>	0.6708±0.1815
3	Accuracy	0.8915±0.1396	0.9429±0.0199	0.7179±0.0831
	Recall	0.6830±0.1024	0.9738±0.0167	0.8305±0.0793
	F1 Score	0.7735±0.0402	<b>0.9580±0.0118</b>	0.7701±0.0395
5	Accuracy	0.8205±0.1732	0.9657±0.0085	0.7956±0.1139
	Recall	0.6415±0.1846	0.9660±0.0104	0.8418±0.1471
	F1 Score	0.7314±0.0279	<b>0.9657±0.0044</b>	0.8108±0.1062
10	Accuracy	0.9153±0.0874	0.9741±0.0031	0.7799±0.1124
	Recall	0.6718±0.1024	0.9662±0.0036	0.8635±0.1190
	F1 Score	0.7749±0.0615	<b>0.9698±0.0015</b>	0.8196±0.0877

TABLE VII: Performance Comparison of Different Feature Generators in StealthLink

N	Evaluation Metric	Feature Generators					
		ResNet-18	ResNet-50	ResNet-101	MLP	LSTM	Transformer
1	Accuracy	0.5772±0.1401	0.6672±0.2501	0.6419±0.0809	0.5996±0.0837	0.5296±0.0332	0.7879±0.1228
	Recall	0.7638±0.2099	0.6759±0.1372	0.7426±0.1843	0.8163±0.2479	0.7127±0.2230	0.8324±0.0484
	F1-Score	0.6575±0.1317	0.6715±0.1004	0.6886±0.0949	0.6914±0.0980	0.6076±0.0845	<b>0.8096±0.0678</b>
3	Accuracy	0.6294±0.1333	0.8205±0.1732	0.6359±0.0772	0.7079±0.0820	0.5642±0.0556	0.9429±0.0199
	Recall	0.7677±0.1387	0.6415±0.1846	0.8511±0.1129	0.7955±0.1009	0.8150±0.1809	0.9738±0.0167
	F1-Score	0.6917±0.1206	0.7314±0.0279	0.7280±0.0301	0.7491±0.0392	0.6668±0.0485	<b>0.9580±0.0118</b>
5	Accuracy	0.6552±0.1733	0.8915±0.1396	0.6933±0.0525	0.8192±0.0663	0.5857±0.0998	0.9657±0.0085
	Recall	0.7584±0.1789	0.6830±0.1024	0.8658±0.0644	0.8528±0.0830	0.7367±0.2023	0.9660±0.0104
	F1-Score	0.7030±0.1607	0.7735±0.0402	0.7688±0.0266	0.8357±0.0488	0.6526±0.0814	<b>0.9657±0.0044</b>
10	Accuracy	0.7381±0.0381	0.9153±0.0874	0.6836±0.0684	0.8760±0.0433	0.6090±0.0830	0.9741±0.0031
	Recall	0.8586±0.0581	0.6718±0.1024	0.8758±0.0608	0.9039±0.0755	0.7837±0.1513	0.9662±0.0036
	F1-Score	0.7938±0.0279	0.7749±0.0615	0.7679±0.0550	0.8897±0.0355	0.6854±0.0282	<b>0.9698±0.0015</b>

TABLE VIII: Performance Comparison of Different Transfer Learning Frameworks in StealthLink

N	Evaluation Metric	Transfer Learning Frameworks			
		DANN	DAN	MCD	w/o Transfer
1	Accuracy	0.7957±0.0691	0.6545±0.0993	0.7879±0.1288	0.5637±0.0675
	Recall	0.6301±0.0726	0.6881±0.1590	0.8324±0.0484	0.7358±0.1470
	F1 Score	0.7033±0.0115	0.6709±0.0536	<b>0.8096±0.0678</b>	0.6383±0.0523
3	Accuracy	0.8483±0.0096	0.8510±0.0094	0.9429±0.0199	0.5275±0.0365
	Recall	0.7347±0.0199	0.7617±0.0370	0.9738±0.0167	0.8106±0.1911
	F1 Score	0.7879±0.0105	0.8041±0.0217	<b>0.9580±0.0118</b>	0.6391±0.0549
5	Accuracy	0.8404±0.0116	0.8770±0.0088	0.9957±0.0085	0.5636±0.0423
	Recall	0.7882±0.0118	0.8299±0.0223	0.9660±0.0104	0.7438±0.1711
	F1 Score	0.8136±0.0080	0.8529±0.0103	<b>0.9806±0.0044</b>	0.6413±0.0339
10	Accuracy	0.8675±0.0109	0.8775±0.0079	0.9741±0.0031	0.5512±0.0799
	Recall	0.8989±0.0032	0.8984±0.0045	0.9662±0.0036	0.7733±0.1871
	F1 Score	0.8829±0.0065	0.8878±0.0047	<b>0.9698±0.0015</b>	0.6436±0.0417

learning frameworks. We speculate that this is because MCD, through the maximum classifier discrepancy strategy with dual classifiers, can effectively leverage domain knowledge from the malicious account detection task, generating clear and robust decision boundaries for classifying coin mixing accounts even when the cross-task feature space is not aligned.

To further evaluate the effectiveness of the transfer learning framework in cross-task knowledge transfer, this section uses the t-SNE dimensionality reduction algorithm to visually compare the high-dimensional latent feature space distributions of the MCD framework and the baseline model without transfer learning (“w/o Transfer”). The experimental data follows the cross-domain balance principle: 450 malicious accounts and 450 normal accounts are randomly selected from the BABD malicious account detection dataset to form the source domain samples, and an equal number of target domain samples are drawn from the Tornado Cash coin mixing transaction dataset. As shown in Figure 5, the blue, yellow, and green legends in the figure represent the latent feature distributions of malicious accounts, normal accounts, and coin mixing samples, respectively.

As illustrated in Fig. 5, the MCD-based transfer learning framework significantly enhances the separability of the coin mixing transaction tracing task, effectively achieving knowledge transfer from the malicious account detection task. Specifically, compared to the baseline model without transfer learning (w/o Transfer), the t-SNE visualization generated by the MCD framework reveals a pronounced spatial separation between clusters of malicious and benign accounts. Furthermore, coin mixing samples exhibit a linearly separable pattern along the main discriminative direction, indicating improved inter-class separability and intra-class compactness. **Classifier Evaluation.** During the model fine-tuning stage, the choice of classifier directly influences the association analysis of mixed transaction address pairs. This section evaluates the impact of seven widely-used classifiers—commonly adopted in blockchain anomaly detection and forensic studies—on the performance of the StealthLink framework. The classifiers include: Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), Logistic Regression (LR), K-Nearest Neighbors (KNN), Random Forest (RF), Long Short-Term Memory network (LSTM), and eXtreme Gradient Boosting (XGBoost). To ensure evaluation reliability, we employ 10-fold cross-validation to systematically test the performance of all classifiers. Detailed experimental results are presented in Table IX.

From Table IX, it can be observed that the MLP classifier is more suitable for identifying the associations among coinjoin-related samples. Specifically, it achieves the highest F1-score under all sample settings. We hypothesize that this is because MLP leverages multiple layers of nonlinear activation functions to extract high-level feature combinations layer by layer, thereby capturing the complex fund flow patterns inherent in coin mixing transactions.

The experimental results indicate that linear classifiers (e.g., Support Vector Machine (SVM), Logistic Regression (LR)) and shallow models (e.g., K-Nearest Neighbors (KNN), Random Forest (RF)) exhibit relatively limited classification

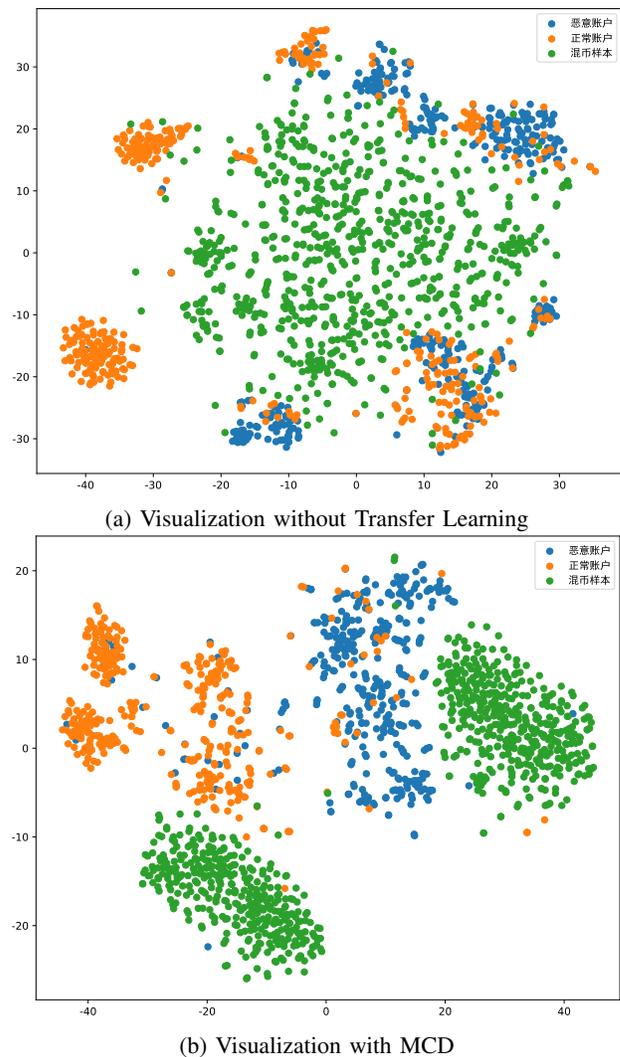


Fig. 5: Visualization of sample embeddings under the transfer learning framework.

performance, with a maximum F1-score of only 67%. We attribute this to the inherent limitations of linear decision boundaries in linear models, which make it difficult to effectively capture nonlinear patterns in coin mixing data—particularly in scenarios involving complex behaviors such as multiple deposits and withdrawals. Meanwhile, shallow classifiers lack the capacity for sufficient feature abstraction, which hampers their ability to perform global feature integration in high-dimensional transaction features.

It is noteworthy that the XGBoost method demonstrates a strong dependence on sufficiently labeled data. When  $N = 1$  or  $N = 3$ , both the accuracy and recall of XGBoost are 0, which is due to the lack of training samples to determine effective splitting points. As the number of samples increases to 5 and 10, XGBoost begins to identify meaningful splits and shows a slight improvement in performance. This result highlights the limitations of tree-based models when dealing with extremely small training datasets.

TABLE IX: Performance Comparison of Different Classifiers in StealthLink Framework

N	Metric	Classifier						
		MLP	SVM	LR	KNN	RF	LSTM	XGBoost
1	Accuracy	0.7879±0.1288	0.5884±0.0711	0.4990±0.0913	0.4990±0.0913	0.5393±0.0948	0.5850±0.1540	0
	Recall	0.8324±0.0484	0.2937±0.1785	0.3585±0.2317	0.3584±0.2316	0.2950±0.1783	0.5371±0.2553	0
	F1 Score	<b>0.8096±0.0678</b>	0.3918±0.0634	0.4172±0.0869	0.4172±0.0868	0.3814±0.0705	0.5600±0.1677	0
3	Accuracy	0.9429±0.0199	0.4630±0.0568	0.4977±0.0548	0.4288±0.0556	0.5219±0.0551	0.6533±0.1095	0
	Recall	0.9738±0.0167	0.3517±0.2707	0.6056±0.3216	0.4899±0.3504	0.6708±0.3023	0.5031±0.2837	0
	F1 Score	<b>0.9580±0.0118</b>	0.3997±0.1640	0.5464±0.1734	0.4674±0.2172	0.5871±0.1680	0.5684±0.2485	0
5	Accuracy	0.9957±0.0085	0.5028±0.0382	0.5461±0.0478	0.4807±0.0506	0.5618±0.0400	0.8647±0.0994	0.5279±0.0421
	Recall	0.9660±0.0104	0.4010±0.2793	0.7407±0.2270	0.5398±0.3242	0.7959±0.1807	0.7330±0.1220	0.6298±0.2823
	F1 Score	<b>0.9806±0.0044</b>	0.4462±0.1609	0.6288±0.1060	0.5085±0.1966	0.6588±0.0911	0.7934±0.1012	0.5279±0.0421
10	Accuracy	0.9741±0.0031	0.5332±0.0266	0.5562±0.0439	0.5312±0.0346	0.5701±0.0310	0.9149±0.0731	0.5218±0.0356
	Recall	0.9662±0.0036	0.4782±0.2714	0.7168±0.2575	0.6667±0.2936	0.8108±0.1649	0.8071±0.0848	0.7142±0.2868
	F1 Score	<b>0.9698±0.0015</b>	0.5043±0.1567	0.6264±0.1323	0.5913±0.1529	0.6695±0.0791	0.8576±0.0782	0.5218±0.0356

## VII. CONCLUSION

In this chapter, we proposed **StealthLink**, a coin-mixing transaction tracing method based on cross-task invariant feature learning. The method introduces a coin-mixing subgraph fusion encoding module to construct an effective joint representation of mixed transactions, and employs a distribution discrepancy minimization strategy to enable knowledge transfer from malicious account detection to the domain of coin-mixing tracing.

Extensive experiments on real-world datasets demonstrate that StealthLink achieves state-of-the-art discrimination performance while exhibiting strong few-shot learning capability and robustness to noisy pseudo-labeled data.

In future work, we plan to extend our approach by incorporating cross-chain tracing mechanisms to explore behavioral patterns of malicious transactions that leverage coin mixing across different blockchains, thereby advancing the tracing of more covert malicious transaction activities.

## REFERENCES

- [1] B. I. for Public Policy, "Crypto, web3 and the metaverse," <https://www.bennettinstitute.cam.ac.uk/wp-content/uploads/2022/03/Policy-brief-Crypto-web3-and-the-metaverse.pdf>, 2022.
- [2] V. Buterin, "A next-generation smart contract and decentralized application platform," <https://github.com/ethereum/wiki/wiki/White-Paper>, 2013.
- [3] T. Cash. (2024) Tornado core. Accessed: 2024-12-09. [Online]. Available: <https://github.com/tornadocash/tornado-core>
- [4] (2024, February) Hackers steal nearly \$10 million from axie infinity co-founder's personal accounts. Accessed on 01/03/2024. [Online]. Available: <https://therecord.media/hackers-steal-millions-from-axie-infinity-founder-personal-accounts>
- [5] (2023, August) Us charges crypto founders over alleged support for north korean hackers. Accessed on 01/03/2024. [Online]. Available: <https://www.aljazeera.com/economy/2023/8/24/us-charges-crypto-founders-over-alleged-support-for-north-korean-hackers>
- [6] F. Béres, I. A. Seres, A. A. Benczúr, and M. Quintyne-Collins, "Blockchain is watching you: Profiling and deanonymizing ethereum users," in *IEEE International Conference on Decentralized Applications and Infrastructures, DAPPS 2021, Online Event, August 23-26, 2021*. IEEE, 2021, pp. 69–78.
- [7] Z. Wang, S. Chaliasos, K. Qin, L. Zhou, L. Gao, P. Berrang, and B. Livshits, "On how zero-knowledge proof blockchain mixers improve, and worsen user privacy," in *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*. ACM, 2023, pp. 2022–2032.
- [8] F. Miedema, K. Lubbertsen, V. Schrama, and R. van Wegberg, "Mixed signals: Analyzing ground-truth data on the users and economics of a bitcoin mixing service," in *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*. USENIX Association, 2023, pp. 751–768.
- [9] L. Wu, Y. Hu, Y. Zhou, H. Wang, X. Luo, Z. Wang, F. Zhang, and K. Ren, "Towards understanding and demystifying bitcoin mixing services," in *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. ACM / IW3C2, 2021, pp. 33–44.
- [10] J. Wu, J. Liu, W. Chen, H. Huang, Z. Zheng, and Y. Zhang, "Detecting mixing services via mining bitcoin transaction network with hybrid motifs," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 52, no. 4, pp. 2237–2249, 2022.
- [11] H. Yang, Z. Li, G. Gou, J. Shi, G. Xiong, and Z. Li, "Bitcoin mixing service detection based on spatio-temporal information representation of transaction graph," in *2023 IEEE International Performance, Computing, and Communications Conference (IPCCC)*. IEEE, 2023, pp. 210–219.
- [12] H. Du, Z. Che, M. Shen, L. Zhu, and J. Hu, "Breaking the anonymity of ethereum mixing services using graph feature learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 616–631, 2024.
- [13] C. Xu, R. Xiong, X. Shen, L. Zhu, and X. Zhang, "How to find a bitcoin mixer : A dual ensemble model for bitcoin mixing service detection," *IEEE Internet of Things Journal*, pp. 1–1, 2023.
- [14] M. Möser, R. Böhme, and D. Breuker, "An inquiry into money laundering tools in the bitcoin ecosystem," in *2013 APWG eCrime Researchers Summit*, 2013, pp. 1–14.
- [15] J. Wu, D. Lin, Q. Fu, S. Yang, T. Chen, Z. Zheng, and B. Song, "Toward understanding asset flows in crypto money laundering through the lenses of ethereum heists," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 1994–2009, 2024.
- [16] Y. Xiang, Y. Lei, D. Bao, T. Li, Q. Yang, W. Liu, W. Ren, and K. R. Choo, "BABB: A bitcoin address behavior dataset for pattern analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 2171–2185, 2024.
- [17] S. Hu, Z. Zhang, B. Luo, S. Lu, B. He, and L. Liu, "BERT4ETH: A pre-trained transformer for ethereum fraud detection," in *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*. ACM, 2023, pp. 2189–2197.
- [18] H. Sun, Z. Liu, S. Wang, and H. Wang, "Adaptive attention-based graph representation learning to detect phishing accounts on the ethereum blockchain," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 3, pp. 2963–2975, 2024.
- [19] W. Chen, Z. Zheng, J. Cui, E. C. H. Ngai, P. Zheng, and Y. Zhou, "Detecting ponzi schemes on ethereum: Towards healthier blockchain technology," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*. ACM, 2018, pp. 1409–1418.
- [20] H. Jin, C. Li, J. Xiao, T. Zhang, X. Dai, and B. Li, "Detecting arbitrage on ethereum through feature fusion and positive-unlabeled learning," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3660–3671, 2022.

- [21] S. Li, G. Gou, C. Liu, C. Hou, Z. Li, and G. Xiong, "TTAGN: temporal transaction aggregation graph network for ethereum phishing scams detection," in *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*. ACM, 2022, pp. 661–669.
- [22] A. Wahrstätter, J. G. Jr., S. Khan, and D. Svetinovic, "Improving cryptocurrency crime detection: Coinjoin community detection approach," *IEEE Trans. Dependable Secur. Comput.*, vol. 20, no. 6, pp. 4946–4956, 2023.
- [23] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, 2012.
- [24] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.
- [25] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.
- [26] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.
- [27] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.
- [28] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017, pp. 1024–1034.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778.
- [30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017, pp. 5998–6008.
- [31] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, ser. JMLR Workshop and Conference Proceedings, F. R. Bach and D. M. Blei, Eds., vol. 37. JMLR.org, 2015, pp. 97–105.
- [32] M. Ghifary, W. B. Kleijn, and M. Zhang, "Domain adaptive neural networks for object recognition," in *PRICAI 2014: Trends in Artificial Intelligence - 13th Pacific Rim International Conference on Artificial Intelligence, Gold Coast, QLD, Australia, December 1-5, 2014. Proceedings*, ser. Lecture Notes in Computer Science, vol. 8862. Springer, 2014, pp. 898–904.
- [33] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 3723–3732.