arXiv:2603.14208v1 [cs.CE] 15 Mar 2026

追踪您的账户：面向隐私保护服务中以太坊的梯度感知动态窗口图框架

Shuyi Miao、Wangjie Qiu、Xiaofan Tu、Yunze Li、Yongxin Wen 和 Zhiming Zheng

摘要——**随着 Web 3.0 技术的快速发展，公有区块链平台上涌现出旨在增强用户隐私和匿名性的新型服务。然而，这些服务固有的强大不可追踪特性，无意间使其成为罪犯洗钱非法资金的理想工具。值得注意的是，现有去匿名化方法在处理此类交易时面临三大挑战：交易语义高度同质化、对时间不连续性的建模能力有限，以及对账户关联图中结构稀疏性的考虑不足。为应对这些挑战，我们提出 GradWATCH，旨在追踪以太坊隐私保护服务中的匿名账户。具体而言，我们首先设计了一个可学习的账户特征映射模块，从原始链上数据中提取有用的交易语义。随后，我们将交易关系纳入账户关联图，以减轻结构稀疏性带来的不利影响。为捕捉时间演变，我们进一步提出一种边感知滑动窗口机制，在三个粒度上传播和更新梯度。最后，我们通过测量学习所得表示空间中的嵌入距离，识别由同一实体控制的账户。实验结果表明，即使在标签不平衡和交易稀疏的情况下，GradWATCH 仍能显著提升性能，平均倒数排名相对提高 1.62% 至 15.22%，F_1 相对提高 3.85% 至 7.31%。**

索引词—**区块链、以太坊、隐私保护服务、动态图神经网络、梯度传播。**

I. 引言

区块链 (BLOCKCHAIN) 技术 [1], 作为加密货币的底层支撑而广为人知，已在金融及其他领域得到广泛应用。然而，在以太坊等公有区块链上，用户使用匿名的字母数字地址进行操作。这种隐匿性使监管监督更加困难，也常被用于非法活动，从而带来安全威胁。为应对这些问题，已有大量研究专注于开发账户去匿名化技术，通过将假名数据与辅助信息 [6]相关联，旨在识别区块链中的个人 [2], [3], 交易 [4],或群体 [5] 。这些现有方法通常通过建模可观察的链上交易路径来学习行为模式，从而实现相对有效的身份识别和行为推断。

Shuyi Miao、Wangjie Qiu、Xiaofan Tu、Yunze Li 和 Zhiming Zheng 隶属于北京航空航天大学北京未来区块链与隐私计算高级创新中心人工智能研究院，地址：中国北京 100191（电子邮箱：shuyimiao@buaa.edu.cn；wangjieqiu@buaa.edu.cn；xiaofantu@buaa.edu.cn；tedlee@buaa.edu.cn；zzheng@pku.edu.cn）。Shuyi Miao、Wangjie Qiu 和 Zhiming Zheng 也隶属于中国北京的中关村实验室。

Yongxin Wen 任职于中国兵器工业第二〇八研究所（中国北京）（电子邮箱：wyx1142557393@163.com）。通讯作者：Wangjie Qiu。

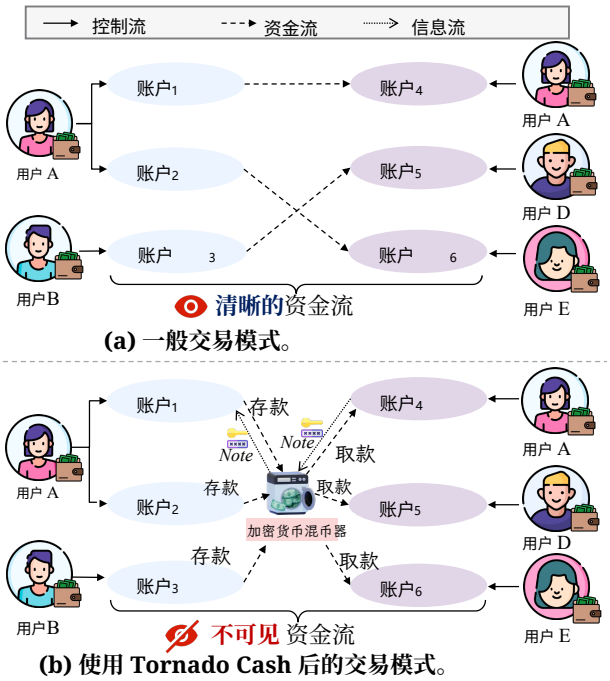


图 1：Tornado Cash 混币机制示意图，对比常规交易与 Tornado Cash 交易，展示存款地址与取款地址如何实现解耦。

然而，Web 3.0 生态系统中的隐私保护技术 [7]，（包括混币服务和隐私增强协议，例如 Railgun）会刻意隐藏或扰乱此类交易路径。Tornado Cash 是一个典型案例，也是以太坊上最早的去中心化混币服务之一。如图 1 所示，Tornado Cash 打破了存款交易与取款交易之间的一一对应关系。在实际操作中，用户A首先使用 $Account_1$向 Tornado Cash 存入固定面额的金额，由此生成与秘密存款凭证相关联的密码学承诺，并将该凭证保留在链下。之后，用户A 可以通过提交根据该凭证生成的零知识证明，授权将资金取至另一个受其控制的账户 $Account_4$ 。重要的是，取款从共享池中执行，且不会引用原始存款，因此在链上消除了确定性的存款与取款关联。这使得资金流无法被直接观察，从根本上削弱了公有区块链上的交易可追踪性。

为提升在非常规交易场景下的隐私保护监管能力，我们聚焦于**账户追踪**任务（即账户关联检测）。这项任务

旨在确定出现在刻意混淆的交易路径中的多个账户是否由同一现实世界实体控制，从而为识别和追踪非法洗钱者提供有效手段。然而，现有的两类去匿名化方案往往难以应对这一任务。**(i) 基于启发式的方法**依赖手工制定规则和领域特定假设，这限制了其可扩展性，也使其难以有效应对刻意逃避检测的对手。**(ii) 基于行为特征的方法** 通过训练预测模型，从区块链数据中学习潜在交易模式，从而自动关联账户。尽管灵活性更高，但基于行为特征的方法在隐私保护环境中仍显不足，并面临若干根本性挑战（C）：**(C1)**同质化交易语义。在混币服务中，由于固定面额和标准化合约交互，交易高度同质化。因此，表层行为特征缺乏足够的判别力，无法捕捉真实的账户控制语义，导致行为相似性成为一种薄弱且充满噪声的信号，难以可靠地进行关联检测。**(C2)** 对时序—结构动态的建模有限。 如图2下半部分所示，基于交易的数据集被划分为50个时间片，各时间片中的账户关联标签分布稀疏且不均匀，并呈现显著变化，某些时期还会突然改变。然而，现有大多数方法依赖静态图建模，而少数动态方法通常将时间信息与图结构割裂开来，因而难以捕捉真实的时序动态。**(C3)** 账户关联图的结构稀疏性。 如图2上半部分所示，只有在时间片10–30期间，账户关联图的图密度才超过 5.32×10^{-7} 的平均值；在大多数时期，其密度都远低于正常图密度标准。这种极端稀疏性严重阻碍了有效的特征传播，并降低了学习稳定性和计算效率。

针对这些挑战，我们提出一种用于隐私保护服务下以太坊账户关联追踪的梯度感知动态窗口图框架（**GradWATcGradWATCH**）。具体而言,-针对**C1**，我们将交易编码为可学习的序列表示，以捕捉时间、金额、空间和上下文属性，并采用重构损失函数确保编码后的特征与原始交易之间的语义对齐。这种设计减轻了交易属性表示高度同质化的倾向，为账户节点提供更可靠的初始表示。其次，针对**C3**，通过分析 Tornado Cash 的机制，我们排除与模糊交易路径相关的其他服务地址，例如中继器和代理。基于净化后的混币交易，我们构建了一个由混币交易驱动的账户关联动态图（MixTAG），从而能够学习交易驱动的账户关联关系，并缓解图稀疏性带来的挑战。第三，针对**C2**，我们提出一种动态异构图框架，将滑动窗口与边感知传播机制相结合。最后，将账户关联表述为

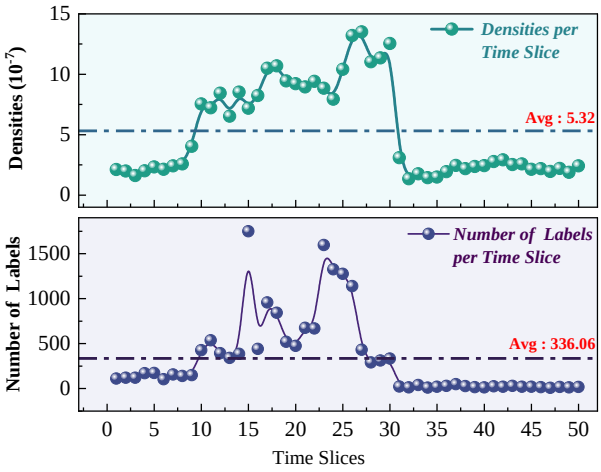


图2：基于交易的数据集（50 个时间片）上动态图每个时间片的密度和标签数量。

链接预测任务，通过测量潜在空间中节点嵌入之间的欧几里得距离 [8]来计算实体控制的可能性。

总而言之，我们的主要**贡献**如下：

- **交易驱动的异构图建模**。我们将账户交易和关联关系整合到一个图中，消除了对专家知识的依赖，同时提高了图密度。
- **端到端账户追踪范式**。 通过将交易到账户映射与边感知窗口机制相结合，我们的方法能够捕获更丰富的交易语义和演化证据，从而促进稳健的账户追踪。
- **实证优势与鲁棒性**。实验结果表明，该方法在两个不同规模的数据集上均表现出色；与基线方法相比，平均倒数排名相对提升了 1.62% 至 15.22%， F_1 提升了 3.85% 至 7.31%。我们的实现已发布于：https://github.com/msy0513/GradWATCH。

II. 相关工作

本节首先定义地址聚类和账户追踪任务，然后详细回顾混币服务中任务检测方面的研究与挑战。最后，介绍动态图链接预测方法的背景和最新进展。

A. 地址聚类与账户追踪

基于公开可用的链上交易，地址聚类 [9]涉及分析账户行为模式，并根据地址是否由同一实体控制，将其划分为尽可能大的子集。当关注点从实体之间的整体控制关系转向判断两个特定账户地址是否属于同一实体时，这项任务称为账户追踪 [10]。账户追踪的核心是识别地址对之间的关联。它可以视为对地址聚类结果的细化与补充

，有助于解决由数据噪声、隐私保护技术或交易模式的复杂性导致的数据不完整或分类错误问题。

B. 混币服务中的账户追踪

在 Tornado Cash 的混币服务中，不同的链上存款地址和取款地址可能对应同一实体，因为持有存款凭证的取款方可以领取资金。由于混币服务会混淆资金流向，许多非法活动通过这些服务隐藏资金来源，进而催生了账户追踪等多种相关下游任务 [10]–[16]。目前针对这项任务的解决方案仍然有限，但现有方案大致可分为两类：**（1）基于启发式的方法**。B´eres 等人 [10] 基于时间活动模式和交易费用模型进行地址追踪。Wang 等人 [11] 提出了一组扩展的启发式方法，利用多面额模式和 TORN 挖矿奖励，进一步揭示零知识混币协议中的地址关联。然而，他们的方法依赖账户关联中固定的行为特征。**（2）基于行为特征的方法**。Hu 等人 [12] 引入了一个用于账户关联检测的预训练 Transformer [17] 。然而，其表现不佳，令人质疑该方法在现实部署中的有效性。与使用语言模型不同，Du 等人 [13] 提出了 MixBroker，它构建了一张账户关联图。该方法高度依赖专家设计特征，而稀疏图结构拓扑有限且缺少时间因素，导致学习效果欠佳、性能不理想。为缓解账户关联稀缺问题，Che 等人 [15] 提出了 StealthLink，这是一种跨任务迁移学习框架，利用恶意账户检测中的不变特征来增强混币追踪能力。Tu 等人[16] 提出了 HiLoMix，它采用频率感知对比学习，并结合基于置信度的监督，以缓解图稀疏性和标签噪声。

C. 动态图链接预测

链接预测是图与网络分析中的一项基础任务，旨在确定一对节点之间是否存在边，或未来是否会形成边 [18]。账户追踪任务可自然地表述为链接预测问题，其目标是预测两个账户节点之间是否存在关联边。为解决链接预测问题，研究人员提出了各种图建模方法，包括图随机游走、静态图神经网络和动态图神经网络。

图随机游走方法（如 DeepWalk [19] 和 Node2Vec [20],）通过模拟游走并捕获结构信息来学习节点嵌入。静态图神经网络方法（包括 GCN [21], 、GAT [22],、GIN [23], 和 Graph- SAGE [24], ）通过邻域聚合在固定图上学习表示，并已证明在链接预测任务中具有良好效果。动态图融入时间信息以建模结构演化。在此背景下，F. Manessi 等人 [25] 提出了 DGNN，该方法使用快照序列和 GRU [26]单元来维护历史状态。A. Pareja 等人 [27] 提出了 EvolveGCN-O 和 EvolveGCN- H，通过演化 GCN 权重或节点表示来建模时间变化。类似地，P. Goyal [28] 提出了 DynGraph2Vec，该方法在每个时间步生成图嵌入，并使用 LSTM[29]捕获时序动态。Zhu 等人 [30]提出了 WinGNN，这是一种无需编码器的框架，通过随机化滑动窗口实现高效的动态图学习。这些方法属于离散时间动态图模型，适用于具有明确时间间隔的图。然而，这些方法主要是通用图建模方法，在面对隐私保护服务下交易网络中常见的结构稀疏性和标签稀缺等挑战时，学习能力有限。因此，亟需开发一种针对动态异构图的账户追踪方法，并融入领域特定的交易特征。

III. 交易驱动的账户关联图构建

本节将详细介绍如何通过融合交易标签和账户关联标签，分两个部分生成 MixTAG：数据处理和图构建。相应流程如图 3 所示。

A. 数据处理

阶段 1：以太坊混币交易数据收集。我们从以太坊区块链浏览器 Etherscan[1] 收集与混币服务合约相关的原始交易数据，这些合约的面额分别为 0.1 以太币、1 以太币、10 以太币和 100 以太币。由于混币交易可能涉及其他服务和合约，例如中继器或代理，一笔混币交易可能包含多笔内部交易。在此过程中，我们保留所有与混币服务相关的外部交易和内部交易。外部交易是由外部拥有账户（EOA）直接发起的链上操作，而内部交易则是在合约账户（CA）之间或合约内部自动执行的消息调用。值得注意的是，内部交易没有自己的交易哈希，而是继承触发它们的外部交易的哈希。

阶段 2：以太坊账户关联标签收集。受先前研究中的启发式标签生成方法启发 [10], [12], [13], ，我们更新并整合了一个现有的带标签数据集。标签来源分为两类：（1）基于混币交易信息的启发式方法：通过分析 gas 价格值的末九位数字中的特定模式，以及存款/取款交易的时间顺序来推断关联关系（例如存款与取款之间的时间间隔）。（2）基于以太坊名称服务（ENS）的启发式方法：通过追踪 ENS 域名转移（经由 setOwner 函数）和子域名创建（经由 setSubnodeOwner 函数）来识别关联关系。例如，如果 $account_1$ 转移了一个 ENS 域名

[1]https://etherscan.io/

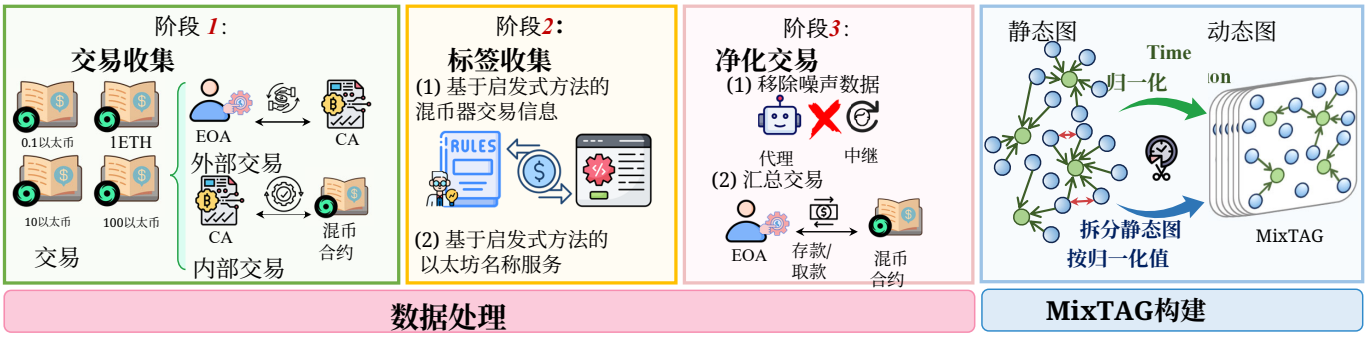


图 3：F 基于混币交易和账户关联标签构建 MixTAG 的流程图 .

所有权给 $account_2$ ，或为账户$account_2$创建了一个子域名，则 $account_1$ 和 $account_2$ 会被标记为由同一实体控制的关联账户。

阶段 3：净化后的混币交易。Tornado Cash 通过提供代理和中继器等辅助服务来增强交易匿名性，使账户能够存入和提取资金，而无需在链上直接与混币合约交互。例如，账户通过代理存款时，会自动触发一笔内部交易，从而避免账户与合约直接交互。同样，在取款时，账户可以同时使用代理和中继器；中继器会预先支付提现金额并收取手续费，因此会产生一笔额外的内部交易。尽管这些服务提升了隐私性，却也大大增加了追踪非法交易（例如与洗钱相关的交易）的难度。将这些额外交易纳入账户关联分析会引入大量噪声，使表征学习和推断变得复杂。为此，我们过滤掉无关的内部交易，仅保留能够直接反映账户与混币合约之间交互的交易，从而构建由内部交易和外部交易组成的“净化”数据集。

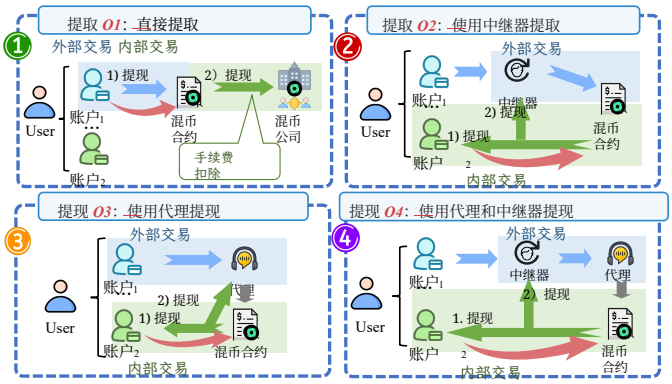


图 4：该图展示了所有可能的提款流程。在图中，*1)* 取款表示将资金取款至账户的交易，而 *2)* 取款表示用于支付提款手续费的交易。粉色箭头表示净化交易中的发送方和接收方。

为实现这一净化过程，我们系统地分析混币服务中所有可能的存款和取款流程，如图 4 和图 5 所示。需要注意的是，我们的数据集仅包含直接涉及混币合约的交易。在某些情况下，例如图 4 中的 (3) 和 (4)，外部交易并未直接关联混币合约，因此只能获得内部交易记录。处理取款交易时，我们首先根据交易哈希是否关联到外部交易对其进行分类。**对于包含外部交易的取款**，如果一笔交易恰好包含两笔内部交易，我们就将其归类为取款，例如图 4 中的 (1) 和 (2)。随后，我们提取实际取款 $account_1$ 及其与混币合约的交互，并将其保留在净化数据集中，标记为取款交易；图 4 中以粉色箭头突出显示了这些交易。**对于不包含外部交易的取款**

，我们将 $account_2$ 识别为提款账户，并将接收方识别为混币合约，从而确保这些交易也被保留，并在图 4 中分别标记为 (3) 和 (4)。这一净化过程消除了无关交互，提高了

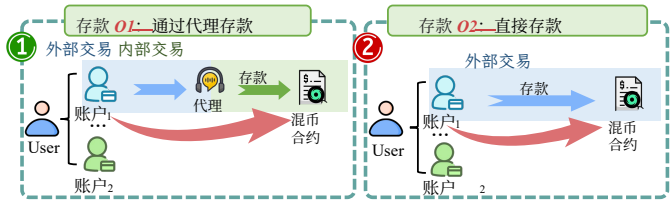


图 5：使用混币服务的所有可能存款流程。粉色箭头表示净化交易中的发送方和接收方。

后续去匿名化分析的准确性。

对于存款交易，如图 5 所示，仅存在两种情形，主要区别在于是否使用代理服务。对于通过代理进行的存款，如图 5（1）所示，我们从内部交易中提取外部交易的发送方（$account_1$）和混币合约信息，将其记录在净化交易数据中，并将其标记为存款交易，如图 5 中的粉色箭头所示。相反，对于直接从 $account_1$ 发起的存款，如图 5（2）所示，我们将对应的外部交易直接记录在净化交易数据中，并同样将其标记为

存款交易。通过上述步骤，得到的净化交易排除了代理和中继服务。此外，通过分析并合并交易，新的数据集明确纳入了每笔交易的发起者及对应的混币合约，从而消除了其他服务带来的噪声和干扰。

*B. MixTAG*构建

本节旨在利用第 III-A 节获得的净化交易构建图。形式化地，我们将由混币交易驱动的净化交易建模为异构图 $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{T}_\mathcal{V}, \mathcal{T}_\mathcal{E}, \mathcal{X}, \mathcal{R}, Y\}$，其中 $\mathcal{V} = \{v_i\}_{i=1}^n$ 表示账户节点的集合，$\mathcal{E} = \{e_i\}_{i=1}^m$ 表示边的集合。$\mathcal{T}_\mathcal{V} = \{v_e, v_c\}$ 表示节点类型的集合，其中 v_e 和 v_c 分别表示外部拥有账户和合约账户。 $\mathcal{T}_\mathcal{E} = \{e_a, e_t\}$ 表示边类型的集合，其中 e_a 和 e_t 分别表示账户关联边和账户交易边。此外，节点 v_i 的类型以及每条边 e_i 均满足 $t_{v_i} \in \mathcal{T}_\mathcal{V}$ 和$t_{e_i} \in \mathcal{T}_\mathcal{E}$。 $\mathcal{X} \in \mathbb{R}^{n \times d_1}$ 为节点特征矩阵，其中 n 为节点数量， d_1 为特征矩阵的维度。$\mathcal{R} \in \mathbb{R}^{m \times d_2}$ 为边特征矩阵，其中 m为边的数量， d_2 为边特征矩阵的维度。我们的任务是确定 MixTAG 中两个节点之间是否存在关联。形式化地，对于节点 v_i 和v_j，若它们由同一个现实世界实体控制，则定义一条边 $e_k = (v_i, v_j)$，并赋予标签 $y_{i,j} = 1$。接着，我们对净化交易中的交易时间 T 进行归一化。具体而言，假设第 k笔混币交易的执行时间为 t_k，则其归一化演化时间定义如下：

$$T_k = \frac{t_k - t_{min}}{t_{max} - t_{min}}, \tag{1}$$

其中， t_{min} 和 t_{max} 分别表示最早和最晚的交易时间。根据交易的演化时间，我们将静态图 $\mathcal{G}$ 划分为 t个时间片。第 k个时间片可表示为 $G_k = \{\mathcal{V}, \mathcal{E}_k, \mathcal{T}_\mathcal{V}, \mathcal{T}_\mathcal{E}, \mathcal{X}, \mathcal{R}_k, T_k, Y_k\}$，其中 $\mathcal{E}_k$不仅包含发生在第 k个时间片内的混币交易边，还包含根据账户提现时间推断出的账户关联边。

IV. 方法

A. 概述

如图 6 所示，GradWATCH 由五个模块组成。(1) 交易到账户映射：该模块处理原始交易，利用数值、时间和位置特征生成初始节点嵌入。重构损失确保嵌入能够准确反映交易数据。(2) 边感知图编码：将其余原始交易细节（不包括模块 (1) 中使用的内容）视为边特征，并应用边感知图卷积网络更新节点表示。(3) 窗口内梯度计算：多个连续的动态时间切片构成一个滑动窗口。在每个窗口内，我们计算当前切片的瞬时梯度、

表一：本文使用的主要符号。

符号	描述
$\mathcal{G}$	MixGraph。
$\mathcal{G}_k$	第 k 个时间切片的 MixGraph。
$v, \mathcal{V}$	节点（账户），节点集
$e, \mathcal{E}$	边（交易或账户关联），边集。
$t_v, \mathcal{T}_\mathcal{V}$	节点类型，节点类型集。
$t_e, \mathcal{T}_\mathcal{E}$	边类型，边类型集。
$x, \mathcal{X}$	节点特征，节点特征矩阵。
$r, \mathcal{R}$	边特征，边特征矩阵。
t, T	交易时间，交易时间集合。
y, Y	账户关联标签，标签集合。
$s_{i,j}, S_i$	第 i 个账户的第 j 个交易序列和集合。
$a_{i,j}$	第 i 个账户的第 j 笔交易中的数值特征账户。
$c_{i,j}$	第 i 个账户的第 j 笔交易中的类别特征账户。
$t_{i,j}$	第 i 个账户中第 j 笔交易的时间特征户。
$p_{i,j}$	第 i 个账户中第 j 笔交易的位置特征。
$H_{i,j}$	第 i 个账户中第 j 笔交易的表示。
$\hat{M}_j$	第 j 个节点收到的消息。
z_i	第 i 个节点的嵌入表示。
Z	节点嵌入集合。
$\nabla \mathcal{L}_{G_t}^{w_i}$	第 i个窗口的瞬时时间切片梯度。
$\nabla \hat{\mathcal{L}}_{G_{t+1}}^{w_i}$	第 i 个窗口的累积时间片梯度。
$\nabla \hat{\mathcal{L}}^{w_i}$	第 i 个窗口梯度。

窗口内的累积梯度以及窗口梯度。(4) 窗口间参数更新：利用窗口梯度和当前窗口最后一个切片的瞬时梯度更新模型参数。这种双层机制能够同时捕捉长期依赖和短期变化，从而实现稳健的更新。(5) 账户关联预测：我们将账户关联检测建模为图链接预测任务，利用更新后的嵌入计算账户节点之间存在边的概率。通过整合这五个模块，GradWATCH 能够有效捕捉混币交易中的动态账户关系，实现高精度链接预测。

B. 交易到账户映射模块

步骤 ***1***：交易表示生成。根据数据特征，交易数据可分为四类：数值、类别、时间和空间。对于每个账户，每笔交易都会被归入相应的数据类型。例如， i个账户的第 j 条交易序列$s_{i,j} = \{a_{i,j}, c_{i,j}, ...\}$的基本特征，例如数值特征和类别特征$a_{i,j}$ 和$c_{i,j}$，可以使用数值编码器进行编码，如下所示：

$$NE(s_{i,j}) = \text{Concat}(W_a \cdot a_{i,j} + b_1, W_c \cdot c_{i,j} + b_2) + \varepsilon, \tag{2}$$

其中， $W_a \in \mathbb{R}^{d_1 \times p}$ 和 $W_c \in \mathbb{R}^{d_2 \times q}$ 分别为类别特征和数值特征的映射矩阵，b_1 和 b_2 为偏置项。为缓解表示中类别之间存在硬边界的问题，并

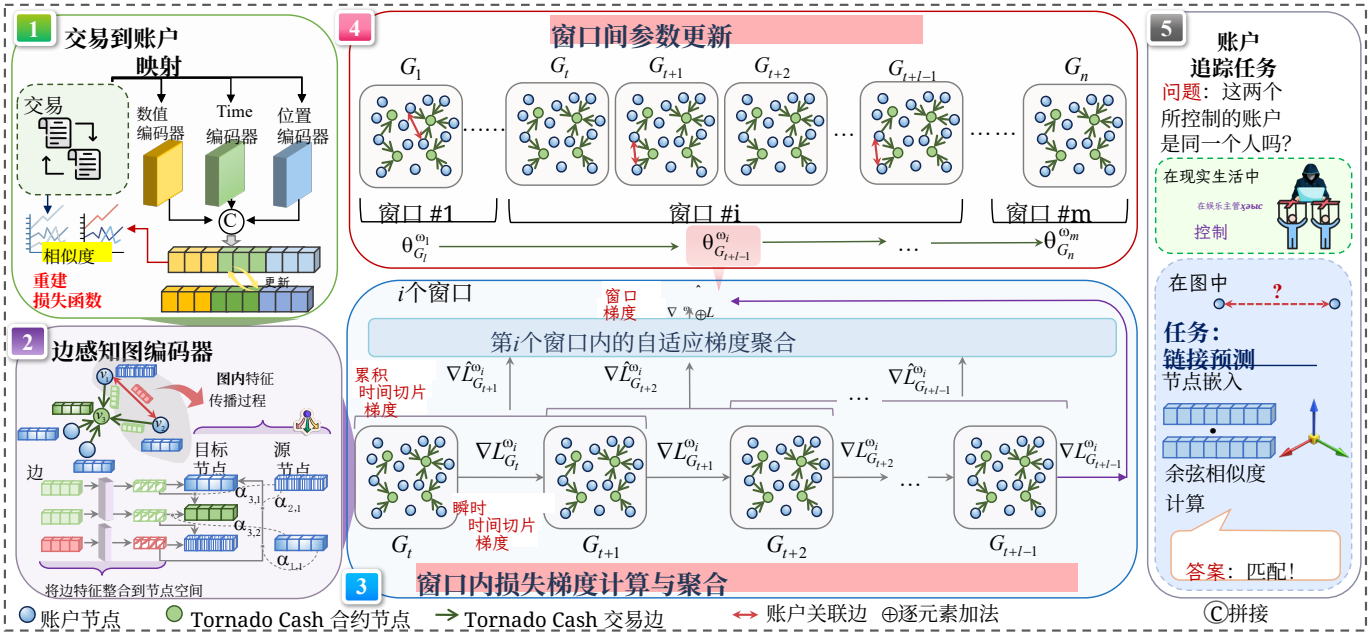


图 6：GradWATCH 概览。

模拟现实中交易数值型数据的变化，我们引入高斯噪声 $\varepsilon \sim \mathcal{N}(0, \sigma^2)$以添加噪声，从而提升数值编码器的鲁棒性和泛化能力。 $d = d_1 + d_2$ 表示使模型能够更好地感知交易的时间信息$t_{i,j}$，我们采用基于正弦和余弦函数的相对时间编码器 [31]–[33]，将时间值映射到高维嵌入空间，使模型能够更好地感知交易时间信息。其表达式如下：

$$TE(s_{i,j}) = \begin{cases} \cos\left(\frac{t_{i,j}}{10000^{(z-1)/d}}\right), & \text{if } z \text{ is odd} \\ \sin\left(\frac{t_{i,j}}{10000^{z/d}}\right), & \text{otherwise} \end{cases} \tag{3}$$

其中， z 表示维度索引，用于递增地生成每个维度的编码。由于不同账户可能与混合服务进行多笔交易，因此，账户表示应根据交易在每个账户交易历史中的相对位置，融入位置编码。具体实现如下：

$$PE(s_{i,j}) = W_p \cdot p_{i,j}, \tag{4}$$

其中， p 是位置索引， $W_p \in \mathbb{R}^{d \times T}$ 表示位置编码矩阵。因此，通过拼接每笔交易的输入值编码器、时间编码器和位置编码器的编码结果，可以得到第j笔交易的综合表示：

$$H_{i,j} = NE(s_{i,j}) \| TE(s_{i,j}) \| PE(s_{i,j}). \tag{5}$$

步骤 ***2***：相似度约束。在生成交易表示后，我们引入基于自编码器的重构损失函数，以优化交易的嵌入。具体而言，我们的重构损失由四个组件构成，每个组件分别对应一个不同的编码器。首先，我们使用交叉熵损失评估交易类别的重构效果，因为它能够有效衡量二元交易类型预测的准确性。其次，对于交易金额编码器，我们使用均方误差（MSE）作为损失函数，以确保学习到的嵌入能够准确反映数值型数据的分布。同样，对于交易位置和时间编码器，我们使用均方误差损失来保留交易的时空信息。

步骤***3***：账户聚合表示。此外，我们对与第 i个账户 v_i相关的所有交易的嵌入进行聚合，第 i个账户的初始表示为：

$$x_i = \frac{1}{|T_i|} \sum_{j \in T_i} H_{i,j}, \tag{6}$$

其中， T_i 是与第 i个账户 v_i 相关的所有交易的集合， $|T_i|$ 是该集合中的交易数量。

C. 边感知图编码

在 MixTAG 中，每个节点 v_i 都具有特征 x_i，并通过边$e_{i,\ j}$与其邻居节点 v_j 的特征 x_j 相连。由于节点与边之间存在维度差异，要在不同空间中实现有效交互与融合颇具挑战。因此，我们使用线性层将边$e_{i,\ j}$ 映射到与节点特征相同的维度，从而获得一个维度相同的新边表示：

$$\tilde{e}_{i,j} = W_e e_{i,j} + b_e, \tag{7}$$

其中， W_e 是权重， b_e 是偏置项。以往研究通常通过图中的边，以相同方式传播和更新节点信息。然而，在 MixTAG 中，除了交易边之外，还存在账户关联

边。因此，有必要针对不同边类型设计不同的消息传播机制。

为了解决这一问题，我们通过整合边特征和注意力机制，改进了 GCN 的传播函数。这一改进使 MixTAG 能够更有效地捕捉涉及多个输入和输出的 MixTAG 中的关键交易上下文信息和账户关联信息。具体而言，我们使用非线性激活函数和线性变换来计算图消息，使消息能够动态分配权重，同时综合考虑源节点 v_i、目标节点 v_j 和边信息$e_{i,\ j}$:

$$m_{i,j} = Sigmoid\left(W_m e_{i,j} + b_m\right) \cdot x_i + \tilde{e}_{i,j} + x_j,$$

其中， W_m 和 b_m 分别为权重和偏置项。接下来，我们聚合目标节点 v_j 接收到的所有消息:

$$M_j = \sum_{i \in \mathcal{N}(j)} m_{i,j}, \tag{8}$$

其中， $\mathcal{N}(j)$表示目标节点 v_j 的邻居集合。鉴于 MixTAG 包含两种边类型：账户关联边和交易边，跨边类型的消息聚合可进一步表示为:

$$\hat{M}_j = \sum_{k \in T_e} M_j^k, \tag{9}$$

其中， T_e 表示所有边类型的集合。最后，我们使用聚合后的消息更新目标节点的节点表示:

$$\hat{x_j} = W_n \cdot \left(\frac{\hat{M}_j}{\|\hat{M}_j\|_2 + \epsilon} \right) + b_n, \tag{10}$$

其中， W_n 和 b_n 分别表示权重和偏置项。 $\|\cdot\|_2$表示 L2 范数， ϵ 是用于确保数值稳定性的常数。

D. 窗口内损失梯度计算与聚合

根据 [30],提出的随机滑动窗口机制，我们将窗口定义为跨越 l个连续时间片的集合。窗口以步长为 s向前滑动，该步长是一个随机变量，也就是说，每个步长都是小于窗口长度 l的随机值。因此，下一个窗口的起始位置必须始终位于前一个窗口内，而其结束位置则延伸至前一个窗口之外。接下来，我们详细介绍窗口内的参数更新机制。受元学习策略 [34], [35], 的启发，我们的方法旨在从多个任务中学习，以快速适应新任务，因此尤其适用于标签稀疏的场景。在每个滑动窗口内，我们不仅计算当前时间片的损失梯度（即**瞬时时间切片梯度**），还累积并计算相邻时间片之间的梯度（即**累积时间片梯度**）。对于整个窗口，我们还会汇总多个累积时间片梯度，以生成窗口梯度，即**窗口梯度**）。

（***1***）瞬时时间切片梯度的计算。动态图 G_t 在时间切片 t经过L 层边感知图卷积编码后，第 i个节点可表示为:

$$h_{i,t}^L = \hat{A} h_{i,t}^{L-1} W_r. \tag{11}$$

接下来，在此时间切片内，使用多层感知机预测账户关联边 betweennode v_i和节点 v_j存在的概率:

$$\hat{y}_{i,j}^t = \text{MLP}(h_{i,t}^L \,||\, h_{j,t}^L), \tag{12}$$

其中，符号 || 表示拼接操作。因此，在一个时间切片内，根据标签数据 $y_{i,\ j}^t \in Y^t$，我们可以使用交叉熵损失计算节点 v_i 和 v_j 之间存在账户关联边的训练损失，如下所示:

$$\mathcal{L}_{G_t}^{w_i} = -\sum y_{i,j}^t \log \hat{y}_{i,j}^t + (1 - y_{i,j}^t) \log(1 - \hat{y}_{i,j}^t). \tag{13}$$

我们将时间切片 G_t 和 G_{t+l}之间的第 i个窗口定义为 w_i，符号 $\mathcal{L}_{G_t}^{w_i}$ 表示第 i个滑动窗口中时间切片G_t 的损失。此外，我们计算对应时间切片的边感知图卷积参数 $\theta_t^{w_i}$的瞬时时间切片梯度，如下所示:

$$\nabla \mathcal{L}_{G_t}^{w_i} \simeq \frac{\partial \mathcal{L}_{G_t}^{w_i}}{\partial \theta_t^{w_i}}. \tag{14}$$

随后，我们使用该梯度更新模型参数，以用于下一个时间切片 G_{t+1}:

$$\theta_{t+1}^{w_i} \longleftarrow (\nabla \mathcal{L}_{G_t}^{w_i})^{\tau} + \theta_t^{w_i}, \tag{15}$$

其中 $\theta_t^{w_i}$为通用学习率。

（***2***）累积时间片梯度的计算。为捕获同一窗口内多个时间片之间的周期性依赖，我们引入了一种累积时间片梯度计算机制，该机制使用当前时间片 G_t 的参数来预测后续时间片 G_{t+1}的结果。 G_{t+1} 上的累积时间片梯度计算如下:

$$\nabla \hat{\mathcal{L}}_{G_{t+1}}^{w_i} \simeq \frac{\partial \mathcal{L}_{G_{t+1}}^{w_i}}{\partial \theta_t^{w_i}}. \tag{16}$$

通过这一迭代过程，当完成第 i 个滑动窗口内从第一个时间片到最后一个时间片的梯度计算后，该窗口总共累积了 $l-1$ 个损失梯度。

（***3***）窗口梯度的计算。这些 l 个损失梯度需要进行自适应聚合，以便对模型参数进行全面调整，如图 7 所示。其形式化表达如下:

$$\nabla \hat{\mathcal{L}}^{w_i} = \phi\left(\nabla \hat{\mathcal{L}}_{G_{t+1}}^{w_i}, \ldots, \nabla \hat{\mathcal{L}}_{G_{t+l-1}}^{w_i}\right), \tag{17}$$

ˆ其中， $\nabla \mathcal{L}^{w_i}$ 表示第 i个窗口梯度， ϕ表示自适应聚合函数。不同于传统的求和方法，为增强动态学习的鲁棒性，我们为窗口内每个时间切片的梯度值分配一个自适应衰减因子。例如，时间$\mathcal{L}_{G_{t+1}}^{w_i}$ 处的累积时间切片梯度 ∇ ˆ

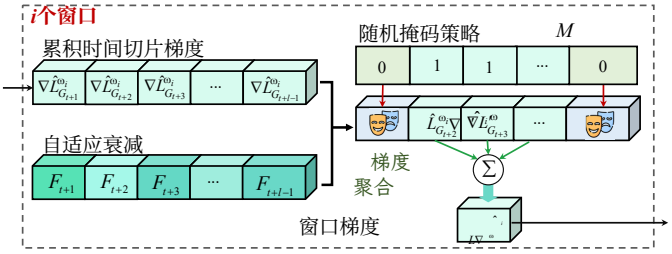


图 7：窗口梯度的计算过程根据累积时间片梯度进行自适应计算。

切片 G_{t+1} 可以分配衰减因子 F_{t+1}，相应的计算可表示为：

$$\nabla\hat{\mathcal{L}}'^{w_i}_{G_{t+1}} = F_{t+1} \odot \nabla\hat{\mathcal{L}}^{w_i}_{G_{t+1}}, \tag{18}$$

$$F_{t+1} = -\frac{\tau}{\sqrt{\delta + r_{t+1}}}, \tag{19}$$

$$r_{t+1} = \rho r_t + (1-\rho)\nabla\hat{L}^{w_i}_{G_{t+1}} \odot \nabla\hat{L}^{w_i}_{G_{t+1}}, r_0 = 0, \tag{20}$$

其中，$\odot$ 表示逐元素乘积运算符，τ 表示模型的一般学习率，δ 表示一个较小的常数，r_t 表示累积梯度项。ρ是用于保留之前时间片梯度权重的平衡项，而（$1-\rho$）表示当前时间片的梯度权重。为进一步减小局部最优解的影响，我们对自适应加权聚合时间片与随机二值掩码执行逐元素乘积，然后将所有结果求和。这将作为第 i 个滑动窗口的最终聚合梯度：

$$\nabla\hat{\mathcal{L}}^{w_i} = \sum_{k=t+1}^{t+l-1} M_k \nabla\hat{\mathcal{L}}'^{w_i}_{G_k}, \tag{21}$$

其中，M 是一个随机二进制掩码矩阵。通过这种方式，GradWATCH 可以从动态图中随机选择一批时间片梯度，并将其聚合为该窗口的综合损失梯度。

E. 窗口间参数更新模块

在窗口之间，通过结合前一个滑动窗口 w_{i+1} 的聚合梯度 $\nabla\mathcal{L}^{w_i}$ 与当前窗口最后一个时间切片 w_i的瞬时时间切片梯度 $\nabla\mathcal{L}^{w_i}_{G_{t+l}}$，更新下一个滑动窗口的图编码参数：

$$\theta^{w_{i+1}}_{t+l-1} \longleftarrow (\nabla\mathcal{L}^{w_i}_{G_{t+l-1}})^{\tau} + (\nabla\hat{\mathcal{L}}^{w_i})^{\tau} + \theta^{w_{i+1}}_{t+l-1}. \tag{22}$$

随着滑动窗口的移动，GradWATCH 的参数会持续更新并传播。最终，我们获得 MixTAG 中所有账户节点的表示，记为 $Z = \{z_1, z_2, ...\}$。

F. 账户关联检测模块

在推理阶段，我们使用经过训练的节点表示来预测所有潜在候选账户连接的链接，包括正链接和负链接。具体而言，我们计算候选集合中两个节点 v_i 和 v_j 的嵌入相似度，以生成该链接存在的概率：

$$\begin{aligned} P(A|Z) &= \prod_{i=1}^{n}\prod_{j=1}^{n} P(A_{ij}|z_i, z_j), \\ P(A_{ij} = 1|z_i, z_j) &= Sigmoid(\theta < z_i, z_j >), \end{aligned} \tag{23}$$

其中，$P(A|Z)$表示节点之间账户关联边的总体概率分布。A 是一个 $n \times n$矩阵，其中每个元素 A_{ij} 表示节点 v_i 与 v_j之间是否存在一条边。Z 是所有节点嵌入的集合。函数 $P(\cdot)$表示节点 v_i 与 v_j之间存在一条边的概率，可以是余弦相似度函数或内积函数。

V. 实验

本节对 GradWATCH 进行了全面实验，旨在回答以下关键问题：

- 研究问题1：实验数据、评估指标和参数是如何配置的?
- 研究问题2：与现有账户追踪方法相比，GradWATCH 的表现如何?
- 研究问题3：GradWATCH 的不同组件对实验结果有何影响?
- 研究问题4：在标注数据稀少时，GradWATCH 的鲁棒性如何？GradWATCH 对参数变化有多敏感?
- 研究问题5：在计算效率方面，GradWATCH 与基线方法相比表现如何?

A. 实验设置（研究问题*1*）

1) 数据集：数据集由交易和标签信息组成。我们从以太坊区块链浏览器 Ether-scan[2]，收集与混币服务相关的原始交易记录，涵盖涉及不同面额（0.1 ETH、1 ETH、10 ETH 和 100 ETH）的混币服务合约的交易。数据集包含 2019-12-16 至 2025-01-06 期间的 526,472 笔交易。经过处理后，我们获得了 150,315 笔净化交易，其中 0.1 ETH 面额的有 57,064 笔，1 ETH 面额的有 11,098 笔，10 ETH 面额的有 11,080 笔，100 ETH 面额的有 71,073 笔。在标注方面，我们采用了受先前研究启发的两种启发式方法 [10], [12], [13]：(1) 基于交易的数据集。(2) 基于 ENS 的数据集。详细说明见第 III-A 节阶段 1，数据集详情见表 II。MixTAG 的密度 D 定义为：$D = {}_{n}{}^{|\mathcal{E}|}{}_{n-1}{}_{()}$，其中，$|\mathcal{E}|$ 表示边的数量，n 表示节点总数。密度值 D 的取值范围为 $[0, 1]$。密度趋近于 0 表示图高度稀疏，而趋近于 1 则表示结构连接紧密。此外，“平均值. 图密度”在

[2]https://etherscan.io/

表 II：账户追踪任务的数据集信息。

数据集	基于交易	基于 ENS
标签数量	16,805	140
时间片数量	100	20
节点数量	107,000	107,000
平均交易边数量	3,508.14	17,540.7
平均账户关联边	168.05	7.00
平均图密度	3.21×10^{-7}	1.53×10^{-6}
平均密度增量	21.9	2,506

表 II 表示所有时间片的平均密度。“平均图密度增幅”表示采用我们的图结构后平均图密度的提升程度，分别为 21.9 倍和 2506 倍。

值得注意的是，尽管两个数据集包含相同数量的账户节点和交易边，但其平均图密度及相应的密度增量却存在显著差异。这些差异主要源于账户关联标签数量的不同，以及所采用的动态图构建策略不同。具体而言，基于交易的数据集包含相对较多的账户关联标签，且这些标签在时间上的分布更为广泛，因此采用固定长度间隔的等时间切片策略。相比之下，基于 ENS 的数据集包含的带标签账户关联少得多，且其时间分布更不均匀，因此采用等数量切片策略，以确保每个时间切片中都有充足的监督信号。因此，在实际应用中，等时间切片可能会产生大量稀疏切片。这些低密度切片会有效稀释平均相对密度变化，因此即使账户关联标签数量较多，整体平均密度增量仍然有限。相比之下，等数量切片会带来更稳定的边分布和更高的平均图密度。引入关联边后，不同切片之间的相对密度波动会更加明显，从而使平均密度增量更大。

2）评估指标：为评估 GradWATCH 在账户追踪任务中的表现，我们采用以下七项指标。ROC 曲线下面积（曲线下面积）用于衡量模型区分正样本和负样本的能力。此外，我们采用五项常见指标：召回率、精确率、F_1、假阳性率和假阴性率。另外，我们使用平均倒数排名来评估模型对正样本的排序性能。平均倒数排名计算所有查询中首个正确预测结果的倒数排名的平均值，从而反映模型排序的有效性，其定义如下：$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$，其中 $|Q|$ 是查询总数，$rank_i$ 表示查询 i 的排序列表中首个被正确预测为正样本的位置。

3）参数设置：GradWATCH 基于 PyTorch 和 DGL 库实现。在将交易级特征转换为账户级特征时，我们将生成的节点嵌入维度设为 544。具体而言，交易类型、交易金额和交易时间均嵌入为 128 维。噪声项为 256 维，位置嵌入为 16 维，投影到节点空间的边特征贡献 14 维。在图编码过程中，我们采用两层边感知图编码器，隐藏层维度为 256，最终节点嵌入维度为 64，丢弃率为 0.1。优化时，初始通用学习率设为 0.02，元学习率为 0.008。经实验确定的最优窗口大小分别为：基于交易的启发式数据集取 15，基于 ENS 的启发式数据集取 5。

B. 对比方法

为全面评估我们提出的方法的性能，我们将基线方法分为三类。**（1）传统机器学习方法**。这类方法包括逻辑回归（LR） [36]和随机森林（RF） [37], 等特征驱动模型, 以及循环神经网络（RNN） [38] 和长短期记忆网络（LSTM） [29], 等用于捕获时间模式的序列建模技术。**（2）图学习方法**。这包括：1）DeepWalk [19] 和 Node2Vec [20], 等图嵌入方法，它们通过随机游走对账户关系进行建模；2）静态图神经网络，包括四种广泛使用的架构：GCN [21], 、GAT [22], 、GIN [23],和 GraphSAGE [24]；以及 3）DGNN [25], 、EvolveGCN [27],、DyGraph2Vec [28],和 WinGNN [30],等动态图学习模型，它们侧重于对账户交互的时间演变进行建模。**（3）近期先进方法**。这类方法无需依赖启发式规则即可完成账户追踪任务，例如 Bert4ETH [12] 和 Mixbroker [13]。此外，我们根据净化交易为每个账户生成 212 维的手工节点特征。这些特征用作基线方法的初始账户特征。更多细节见表 III。

C. 实验结果分析（*RQ2*）

在账户追踪任务中，我们在两个规模各异的启发式数据集（基于交易和基于 ENS）上评估 GradWATCH，以验证其在不同图密度和标签数量下的性能。如表 IV 和 V 所示，GradWATCH 在全部六项关键指标上均取得了最先进的结果。

在基于交易的启发式数据集（图密度高、标签丰富）上，GradWATCH 的曲线下面积达到 97.07%，显著优于所有基线方法。其 F1 分数也达到 89.63%，在精确率（94.91%）和召回率（84.96%）之间展现出良好的平衡。实验结果表明，传统机器学习方法和静态图学习方法往往只能在精确率或召回率方面表现出色，却难以兼顾二者。现有的账户追踪方法（如 Bert4ETH）虽然召回率较高，但精确率较低。这一局限源于其基于行为相似性的检索策略：在高度同质化的交易环境中，这些策略会生成过于宽泛的候选集

表 III：账户的手工特征说明（d：存款，w：取款，a：全部，t：总计，x：阶段索引）。

特征类别	特征数量	特征名称及描述
数值特征（32维）	3 * 5	{0.1、1、10、100、全部} num {d、w、a}：不同用途和不同金额的交易数量。
	2 * 3	{平均值、标准差} num {d、w、a}：不同用途的交易数量的平均值和标准差。
	2 * 2	{金额、平均金额}_{d、w}：不同用途的交易金额总和及平均值。
	2	存款/取款金额、平均存款/取款金额比率：存款/取款交易量及平均值。
	5	d/w 数量 {0.1、1、10、100、a}：不同用途的存款与取款交易比率。
Gas 费用特征（45维）	3 * 3 * 5	gf_{最小值、最大值、平均值} {d、w、a} {0.1、1、10、100、a}：按用途和金额统计的 Gas 费用最大值、最小值及全部费用。
时间特征（90维）	2 * 3 * 5	交易时间 {最早、最晚、l} {d，w，a} {0.1、1、10、100，a}：按用途和金额划分的最早与最晚交易时间。
	4 * 3 * 5	时间间隔_{最小值、最大值、平均值，t} {d、w，a} {0.1、1、10、100，a}：按用途和金额划分的时间间隔。
交叉特征（54维）	3 * 5	交易频率 {d、w，a} {0.1、1、10、100，a}：按用途和金额划分的交易频率。
	3	gas 效率 {d、w、a}：按用途划分的 Gas 费用与交易金额比率。
	12 * 3	bin_x {d、w、a}：按用途和时间段划分的交易计数（$x \in [1, 12]$）。

表四：在基于交易的启发式数据集上，跨多个指标根据标准差对 GradWATCH 和基线方法进行性能评估。**最佳方法**和第二佳方法已标出。（↑）/（↓）表示指标越大越好或越小越好。

方法	基于交易的启发式数据集						
	曲线下面积（↑）	召回率（↑）	精确率（↑）	F_1（↑）	平均倒数排名（↑）	假阳性率（↓）	假阴性率（↓）
逻辑回归 [36]	82.97 ± 1.36	58.21 ± 2.02	81.72 ± 2.05	67.98 ± 1.86	43.04 ± 1.60	18.04 ± 1.30	41.79 ± 2.02
随机森林 [37]	$\underline{83.98 \pm 0.47}$	80.80 ± 1.37	83.81 ± 1.31	$\underline{82.32 \pm 0.61}$	47.57 ± 1.68	17.58 ± 1.68	$\underline{9.20 \pm 1.37}$
循环神经网络 [38]	52.95 ± 1.77	30.28 ± 4.67	52.94 ± 2.16	36.50 ± 9.42	62.15 ± 2.67	27.81 ± 1.59	69.72 ± 4.67
长短期记忆网络 [29]	51.75 ± 1.01	49.65 ± 9.13	52.49 ± 2.43	48.83 ± 9.52	50.00 ± 1.09	46.07 ± 9.56	50.35 ± 9.13
DeepWalk [19]	60.27 ± 0.73	59.98 ± 0.84	79.75 ± 4.58	68.47 ± 1.26	52.30 ± 3.03	18.85 ± 1.09	33.47 ± 0.83
Node2Vec [20]	62.91 ± 2.37	61.32 ± 0.81	81.85 ± 3.83	70.12 ± 1.36	58.21 ± 1.64	19.10 ± 0.44	39.17 ± 0.81
图卷积网络 [21]	63.49 ± 1.59	76.83 ± 1.78	53.19 ± 1.90	62.83 ± 1.32	59.84 ± 1.02	67.71 ± 4.13	23.16 ± 1.78
图注意力网络 [22]	70.03 ± 3.42	83.35 ± 5.35	54.35 ± 1.70	66.67 ± 2.69	60.07 ± 2.17	72.44 ± 3.49	13.65 ± 5.35
图同构网络 [23]	66.81 ± 1.52	26.24 ± 1.07	$\underline{92.49 \pm 2.37}$	40.85 ± 1.15	64.25 ± 1.01	$\underline{17.57 \pm 0.82}$	73.73 ± 1.07
GraphSAGE [24]	67.90 ± 2.04	79.91 ± 5.42	56.55 ± 2.38	66.06 ± 1.31	69.10 ± 1.02	61.91 ± 8.90	20.09 ± 5.42
动态图神经网络 [25]	79.94 ± 3.79	60.00 ± 2.13	52.03 ± 1.90	55.01 ± 1.89	54.77 ± 2.17	35.69 ± 2.91	41.00 ± 2.14
EvolveGCN-O [27]	61.42 ± 9.82	70.62 ± 5.35	72.55 ± 8.05	64.12 ± 3.89	62.60 ± 0.21	47.78 ± 9.28	29.38 ± 2.35
EvolveGCN-H [27]	80.74 ± 1.19	83.54 ± 4.73	78.08 ± 3.36	81.79 ± 0.83	62.60 ± 0.11	25.06 ± 6.40	13.46 ± 4.73
Dygraph2vec [28]	83.65 ± 1.07	60.90 ± 3.46	91.13 ± 1.41	71.86 ± 1.57	$\underline{75.99 \pm 0.84}$	17.58 ± 3.71	39.10 ± 2.33
WinGNN [30]	80.36 ± 1.37	74.51 ± 3.11	76.56 ± 1.85	75.52 ± 2.33	73.91 ± 0.93	22.80 ± 3.71	25.49 ± 2.33
Bert4ETH [12]	63.20 ± 1.07	$\mathbf{89.13 \pm 1.07}$	50.93 ± 1.07	64.82 ± 1.07	32.51 ± 1.07	84.57 ± 1.07	10.87 ± 1.07
Mixbroker [13]	56.37 ± 0.56	38.05 ± 0.64	60.08 ± 1.04	46.59 ± 0.62	53.27 ± 2.34	25.31 ± 1.09	61.95 ± 0.64
GradWATCH	$\mathbf{97.07 \pm 0.76}$	$\underline{84.96 \pm 2.32}$	$\mathbf{94.91 \pm 3.06}$	$\mathbf{89.63 \pm 2.17}$	$\mathbf{91.21 \pm 2.95}$	$\mathbf{17.33 \pm 3.00}$	$\mathbf{7.75 \pm 1.17}$
%提升。	13。09%	-4.17%	2.42%	7.31%	15。22%	0.24%	1。45%

，从而导致大量假阳性匹配。相比之下，GradWATCH 在动态图中同时确保结构和时间的一致性，仅在跨时间窗口观察到充分且一致的证据时才预测控制链接。这种设计有效减少了假阳性，并提高了检测可靠性。因此，我们的方法平均倒数排名达到 91.21%，较表现最佳的竞争方法提升了 15.22%。

在基于 ENS 的启发式数据集（低图密度、稀疏标签）上，随着图稀疏性增加、标签可用性降低，所有方法的性能均有所下降。然而，如表 V 所示，GradWATCH 在全部五项评估指标上始终优于所有基线方法，尤其在曲线下面积（84.26%）和平均倒数排名（68.22%）等关键指标上，分别领先第二名方法 10.95% 和 1.62%。这些结果表明，GradWATCH 通过其图构建策略缓解了图稀疏性，而具备边感知能力的窗口梯度传播机制则提供了必要的时间推理信号，使其即使在稀疏动态图和有限监督条件下也能保持稳健的性能。我们进一步分析 GradWATCH 未能在任一数据集上取得最高召回率的原因。首先，这一结果反映了精确率与召回率之间固有的权衡。其次，由于受控账户关系呈长尾分布，当相关样本分布在多个时间窗口中时，滑动窗口建模方法可能无法汇聚足够的上下文信息。

总而言之，尽管我们的方法无法保证在全部七项评估指标上都达到最优表现，但总体而言，它能更准确、更稳定地进行账户关联检测，尤其是在具有挑战性的情

况下表现出卓越的鲁棒性，例如交易稀疏且标签高度不平衡的情况。

D. 消融实验（研究问题*3*）

研究问题*2*证明了我们提出的 GradWATCH 在账户追踪任务中具有强大的推理能力。为进一步验证精心设计的组件的有效性，我们通过依次移除或替换关键模块或机制，并评估其对整体性能的影响，开展了消融研究。具体而言，*"- w/oEdge-Aware GraphEnc."* 表示移除消息传递过程中的边感知图编码器这一消融设置。因此，边所携带的原始交易信息会被丢弃，模型转而采用传统的基于 GCN 的传播机制。*"-w/o T.-to-A. Map."*表示移除交易到账户映射模块，并使用随机初始化的节点表示代替映射嵌入。*"-w/oMixTAG"* 表示仅使用账户关联边构建同质图，同时排除所有其他类型的交互。*"-w/o Intra-Win.Grad Com."*变体会移除窗口内各时间片之间的梯度计算与传播。*"-w/o Window"* 则直接将窗口大小固定为单个演化帧。需要注意的是，调整窗口大小的实验结果见超参数敏感性分析。

如表 VI 所示，消融实验结果表明，移除任何模块都会导致性能下降。具体而言，同时移除窗口以及窗口内损失梯度计算与聚合模块，会使曲线下面积下降 24.22%，F_1下降 21.30%。保留窗口，但删除窗口内的梯度计算与传播后，曲线下面积和 F_1 分别提升 14.82% 和 12.15%。然而，在全部四项指标上，这种配置的表现仍比我们的模型低 6.72% 至 9.92%。此外，移除 MixTAG 会使召回率和精确率分别下降 7.41% 和 16.50%，证明了采用混币交易驱动的账户推断图结构的必要性。最后，移除边感知编码器和交易—账户映射模块后，全部四项指标的表现均有所下降。这表明，利用原始交易信息、采用异构边传播与聚合机制，并从原始交易中提取高阶账户特征，对于提升隐私保护服务下的账户追踪性能具有很强的针对性和有效性。

E. 模型鲁棒性（研究问题*4*）

在评估模型的整体性能后，我们研究了超参数设置的影响，以验证所提出方法的鲁棒性。此外，我们提出以下**观察**与可解释性见解。

1) 观察 ***1***：用于图构建和监督的标签之间存在最佳平衡。受结构学习理论 [39] 和解耦表示原则的指导

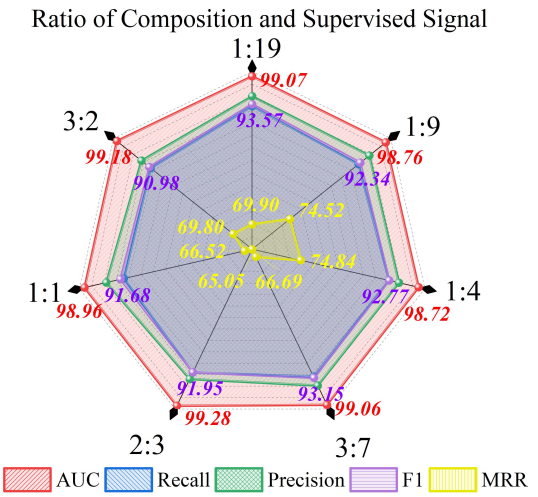


图8：探索基于交易的启发式数据集上用于图构建和监督的标签之间的最佳平衡。

[40], 我们构建了一个双路径标签解耦框架，在账户追踪任务中强制区分"是什么"（监督信号）和"怎么做"（图构建）。具体而言，原始标签被分为两类：i）用于二元分类的监督信号；ii）用于在图中构建账户间约束边的组合元素。如图 8 所示，实证结果表明，当组合元素与监督信号的比例为 1:4 时，模型性能达到最佳（F_1 = 92.77%，平均倒数排名 = 74.84%），揭示了结构化先验与监督目标之间的动态平衡机制。这种平衡表明，适度的关系约束有助于提升对复杂交易模式的推理能力，而过度的结构学习则会引入与任务无关的特征干扰。

2）观察***2***：标签有限时模型性能仍保持稳定。在现实场景中，标签通常较为稀缺。为模拟这一情况，我们调整整个数据集中训练样本和测试样本的比例，实验结果如图9所示。具体而言，我们将训练集比例从20%调整至60%，相应地将测试集比例从80%调整至40%。结果显示，曲线下面积、召回率、精确率和 F_1等核心指标始终保持在90%左右，表明模型具有较强的韧性。值得注意的是，当训练比例超过30%时，平均倒数排名指标显著提升。然而，即使训练数据仅占20%，平均倒数排名仍高于70%。这些结果表明，减少带标签的训练数据对整体性能的影响微乎其微，证实了所提出模型的鲁棒性。

3）观察***3***：超参数变化下的模型鲁棒性。为了研究模型对超参数的敏感性，我们选取了两个关键超参数——时间片数量和滑动窗口大小，评估它们对性能的影响。如图10a所示，在基于混币交易的数据集上进行调优时，将时间片数量从40增加到100，所有评估指标均持续提升。这表明，100个时间

表 V：GradWATCH 与基线方法在基于 ENS 的启发式数据集上多个指标的标准差性能评估。**最佳** 和次佳结果已标出。(↑) / (↓) 表示该指标越优的方向。

方法	基于 ENS 的启发式数据集						
	曲线下面积 (↑)	召回率 (↑)	精确率 (↑)	F_1 (↑)	平均倒数排名 (↑)	假阳性率 (↓)	假阴性率 (↓)
逻辑回归 [36]	58.89 ± 1.97	43.36 ± 2.35	63.07 ± 2.90	49.98 ± 1.82	24.55 ± 3.21	24.55 ± 3.21	56.64 ± 1.35
随机森林 [37]	61.67 ± 1.64	56.67 ± 2.00	59.53 ± 1.33	58.06 ± 1.88	22.48 ± 5.44	22.48 ± 5.44	15.33 ± 1.00
循环神经网络 [38]	48.89 ± 2.89	58.83 ± 1.73	53.36 ± 2.97	52.10 ± 1.82	52.76 ± 2.89	62.76 ± 2.89	41.17 ± 1.73
长短期记忆网络 [29]	40.00 ± 3.89	24.19 ± 3.79	23.62 ± 2.96	21.72 ± 2.98	50.00 ± 3.24	39.73 ± 2.42	75.81 ± 2.79
图卷积网络 [21]	63.14 ± 2.24	27.31 ± 2.17	53.05 ± 4.29	34.07 ± 3.84	26.38 ± 2.98	29.5 ± 3.50	72.69 ± 2.17
图注意力网络 [22]	66.18 ± 1.27	40.00 ± 3.39	62.31 ± 2.89	48.72 ± 1.31	27.09 ± 3.27	28.48 ± 5.53	60.00 ± 3.39
图同构网络 [23]	65.60 ± 8.25	27.14 ± 8.25	55.14 ± 3.61	34.03 ± 2.75	25.46 ± 3.63	23.15 ± 11.37	7.86 ± 8.25
动态图神经网络 [25]	73.31 ± 1.85	60.10 ± 2.12	52.03 ± 2.01	55.01 ± 1.89	42.96 ± 3.31	55.69 ± 3.77	40.00 ± 2.12
GraphSAGE [24]	54.51 ± 2.71	79.73 ± 5.73	50.54 ± 6.06	61.01 ± 6.52	22.66 ± 7.10	77.14 ± 8.66	20.27 ± 5.73
EvolveGCN-O [27]	54.32 ± 4.61	68.43 ± 9.56	60.64 ± 3.73	53.79 ± 3.57	24.03 ± 0.01	59.79 ± 6.23	31.57 ± 9.56
EvolveGCN-H [27]	69.67 ± 5.88	76.42 ± 9.36	55.58 ± 5.86	64.37 ± 5.27	24.03 ± 0.14	47.08 ± 4.09	15.58 ± 9.36
Dygraph2vec [28]	69.69 ± 1.23	71.00 ± 0.11	56.01 ± 0.45	62.62 ± 0.23	32.13 ± 2.12	25.01 ± 4.08	20.00 ± 3.26
WinGNN [30]	70.45 ± 1.15	71.85 ± 0.15	56.88 ± 0.42	63.50 ± 0.25	33.02 ± 2.05	24.15 ± 3.95	19.20 ± 3.10
Bert4ETH [12]	59.66 ± 1.27	**79.83 ± 3.21**	49.45 ± 2.01	61.06 ± 1.68	39.08 ± 3.25	**21.34 ± 1.07**	21.71 ± 2.46
Mixbroker [13]	50.93 ± 9.21	76.67 ± 4.68	51.69 ± 5.61	61.76 ± 2.98	20.83 ± 1.94	83.33 ± 9.72	15.33 ± 4.68
GradWATCH	**84.26 ± 1.45**	64.95 ± 3.84	**71.77 ± 2.91**	**68.22 ± 2.32**	**54.38 ± 5.26**	26.76±11.75	**14.45 ± 9.93**
%改进。	10.95%	−14.88%	8.70%	3.85%	1.62%	−5.42%	0.88%

表 VI：基于交易的启发式数据集上不同模型变体的性能比较。

模型变体	AUC	召回率	精确率	F_1
GradWATCH	**97.07**	**84.96**	**94.91**	**89.63**
- 去除 边缘感知图编码	94.49	78.86	88.36	80.34
-去除 时间到属性映射	92.67	81.10	83.97	79.91
-去除 MixTAG	89.66	79.30	78.41	78.84
- 去除 窗口内梯度合并	87.67	78.24	84.99	81.48
-无窗口	72.85	78.78	61.91	69.33

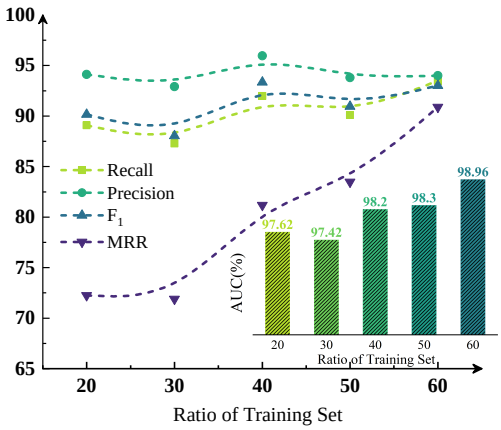
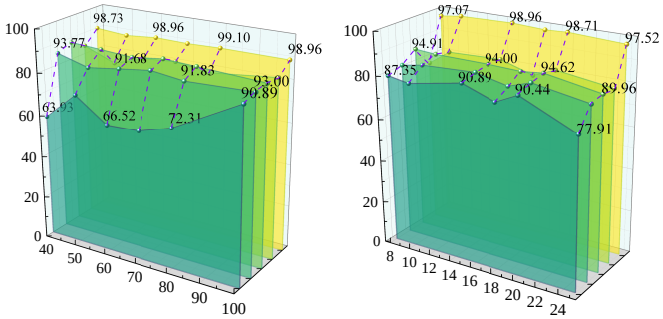


图9：训练集比例对基于交易的启发式数据集性能的影响。

片是该数据集的最佳设置。此外，在整个调优过程中，性能保持相对稳定，这表明我们的方法对变化具有鲁棒性



(a) 时间切片的影响。 (b) 窗口大小的影响。

图10：在基于交易的启发式数据集上调整时间切片和窗口大小。

时间片的数量。

接下来，我们将时间片数量固定为100，并调整窗口大小，以探索最优配置及其对实验性能的影响。如图10b所示，随着窗口大小从8增加到25，模型性能先提升后下降；窗口大小设为15时，性能达到最高。窗口大小为1时的设置结果，列于V-D中标记为 *"-w/o Window"* 的消融设置下。这些结果表明，窗口机制对模型性能具有重要作用，而调整过程中的波动仍处于相对较小的范围内。

F. 计算效率（研究问题*5*）

在本小节中，我们通过理论复杂度分析和实验评估，研究 GradWATCH 的计算效率。

表VII：基于交易的启发式数据集上的平均每轮运行时间、峰值内存使用量和推理吞吐量结果。

	时间（秒）(↓)			内存（GB）(↓)		吞吐量（样本/秒）(↑)
方法	预处理时间	训练时间	推理时间	峰值 RAM	峰值 GPU	推理吞吐量
EvolveGCN-H	110.31	**42.36**	**0.73**	29.37	7.36	9208
Dygraph2vec	110.31	56.45	0.78	35.62	8.37	8618
WinGNN	110.31	44.23	0.74	29.92	6.29	9084
Bert4ETH	113.99	2,670.91	2154.48	7.01	16.41	3.12
Mixbroker	100.32	29.7	0.57	**1.06**	**0.29**	**11793**
GradWATCH	**90.89**	62.08	0.74	30.93	6.50	9084

1) 时间与空间复杂度的理论分析。GradWATCH 的运行时间主要由两个部分组成：(i) 交易到账户映射；(ii) 边感知滑动窗口编码。设 n 为账户节点数，T 为时间片数。我们用 $\bar{m}$ 表示每个时间片的平均交易边数，d_h 表示隐藏维度，d_1 表示构造的账户特征维度。映射步骤的时间复杂度为 $\mathcal{O}(N_{tx} \cdot d_1)$，其中 N_{tx} 表示用于构造账户特征的交易数。在编码过程中，GradWATCH 在 T 个时间片上执行消息传递，时间复杂度为 $\mathcal{O}(T \cdot (\bar{m} \cdot d_h + n \cdot {d_h}^2))$。该过程通常由区块链图中的逐边聚合主导，可近似表示为 $\mathcal{O}(T \cdot \bar{m} \cdot d_h)$。值得注意的是，计算仅针对观测到的交易边进行，并使用稀疏图算子，从而避免对所有节点对进行密集的成对交互。

就空间复杂度而言，GradWATCH 存储 T 个时间图及其特征。图结构中，交易边需要 $\mathcal{O}(T \cdot \bar{m})$ 的内存；如果包含关联边，则需要 $\mathcal{O}(T \cdot m_a)$，其中 m_a 表示每个时间切片的平均关联边数量。账户节点特征的存储成本为 $\mathcal{O}(T \cdot n \cdot d_1)$，维度为 d_2 的边特征存储成本为 $\mathcal{O}(T \cdot \bar{m} \cdot d_2)$。对于一个 L 层编码器，模型参数的规模为 $\mathcal{O}(L \cdot {d_1}^2)$，而训练时激活值还会额外占用 $\mathcal{O}(L \cdot (n \cdot d_1 + \bar{m} \cdot d_1))$ 的内存。总体而言，主要内存开销随存储的快照数量和稀疏边数量线性增长。

2）效率实验分析。如表VII所示，我们报告了 Grad-WATCH 和若干性能强劲的对比方法的运行时间、峰值内存使用量（包括 RAM 和 GPU内存）以及推理吞吐量。表中的时间相关指标包括平均处理时间、训练时间和每轮次推理时间。吞吐量在测试集上测量，并以每秒样本数（samples/s）报告。从结果可以看出，MixBroker 得益于其简洁的建模方法设计，整体效率较高，但实验性能相对较差。Bert4ETH 将每个账户地址映射为词典中的一个词元，导致词汇表规模急剧膨胀，进而产生特别高的时间开销。传统动态图处理方法在预处理期间需要耗费大量时间进行特征工程，以构建最优的初始账户特征。相比之下，我们的端到端方法针对账户关联引入了任务特定优化策略，并以可接受的计算开销增幅实现了更优性能。

VI. 结论

随着区块链技术的快速发展，地址追踪已成为合规与反洗钱工作中的核心研究重点。本文提出了 GradWATCH，一种新颖的梯度感知动态窗口图学习框架。具体而言，GradWATCH 将涉及混币服务的交易建模为动态异构图。它采用滑动窗口机制，在每个窗口内跨时间切片传播边感知图编码器的梯度，并自适应地聚合这些瞬时梯度，形成累积梯度和窗口梯度。随后，将这些梯度整合起来以更新模型参数，从而生成稳健的账户表示。最后，该方法通过衡量学习得到的表示之间的相似度，判断两个地址是否对应同一现实世界实体。在多个采用启发式规则生成、规模各异的数据集上进行的实验结果表明，GradWATCH 始终取得最先进的性能。未来工作将致力于提升模型在跨链场景中的泛化能力，并应对不断演变的混币协议以及通过跨链桥实施的嵌套洗钱策略所带来的挑战。

VII. 致谢

本研究得到北京市自然科学基金（项目编号：Z230001）、国家自然科学基金（项目编号：62572029）、北航甘伟行动计划（项目编号：JKF-2025083360604）、国家网络安全科技重大专项（项目编号：2025ZD1501002）、北京未来区块链与隐私计算高精尖创新中心以及国家留学基金管理委员会（CSC）的资助。

参考文献

[1]D. Ressi、R. Romanello、C. Piazza 和 S. Rossi，“人工智能增强的区块链技术：进展与机遇综述”，《网络与计算机应用期刊》，第 103858 页，2024年。[2] S. Miao、W. Qiu、H. Zheng、Q. Zhang、X. Tu、X. Liu、Y. Liu、J. Dong 和 Z. Zhiming，“了解你的账户：基于双图推断的以太坊账户去匿名化”，2025 IEEE第41届国际数据工程会议（ICDE），2025年。

[3] D. Lin、J. Wu、T. Huang、K. Lin 和 Z. Zheng，“以太坊上谁是谁？使用异配图卷积网络进行账户标注”，IEEE《系统、人类与控制论：系统》汇刊，第 54 卷，第 3 期，第 1541–1553 页，2023年。[4] B. Han、Y. Wei、Q. Wang、F. M. D. Collibus 和 C. J. Tessone，“Mt 2 ad：使用图神经网络在以太坊网络中进行多层时序交易异常检测”，《复杂与智能系统》，第 10 卷，第 1 期，第 613–626 页，2024年。[5] J. Liu、J. Chen、J. Wu、Z. Wu、J. Fang 和 Z. Zheng，“钓出欺诈者：利用区块链数据揭露以太坊钓鱼团伙”，IEEE《信息取证与安全》汇刊，第 19 卷，第 3038–3050 页，2024 年。[6] Q. Feng、D. He、S. Zeadally、M. K. Khan 和 N. Kumar，“区块链系统中的隐私保护综述”，《网络与计算机应用期刊》，第 126 卷，第 45–58 页，2019年。[7] J. J. Si、T. Sharma 和 K. Y. Wang，“理解 Web3生态系统中的用户感知的安全风险及缓解策略”，载于《2024 CHI计算系统人因大会论文集》，2024年，第 1–22 页。[8] P.-E. Danielsson，“欧氏距离映射”，《计算机图形学与图像处理》，第 14 卷，第 3 期，第 227–248 页，1980年。[9] F. Victor，“以太坊地址聚类启发式方法”，载于《金融密码学与数据安全：第24届国际会议，FC 2020，马来西亚亚庇，2020年2月10日至14日修订精选论文24》，Springer，2020年，第 617–633 页。[10] F. B´eres、I. A. Seres、A. A. Bencz´ur 和 M. Quintyne-Collins，“区块链正在监视你：以太坊用户画像与去匿名化”，载于《2021 IEEE去中心化应用与基础设施国际会议（DAPPS）论文集》。IEEE，2021年，第 69–78 页。[11] Z. Wang、S. Chaliasos、K. Qin、L. Zhou、L. Gao、P. Berrang、B. Livshits 和 A. Gervais，“零知识证明区块链混币器如何改善及损害用户隐私”，载于《ACM Web大会2023论文集》，2023年，第 2022–2032 页。[12] S. Hu、Z. Zhang、B. Luo、S. Lu、B. He 和 L. Liu，“Bert4ETH：用于以太坊欺诈检测的预训练 Transformer”，载于《ACM Web大会2023论文集》，2023年，第 2189–2197 页。[13] H. Du、Z. Che、M. Shen、L. Zhu 和 J. Hu，“利用图特征学习破解以太坊混币服务的匿名性”，IEEE《信息取证与安全》汇刊，第 19 卷，第 616–631 页，2024年。[14] M. Wu、W. McTighe、K. Wang、I. A. Seres、N. Bax、M. Puebla、M. Mendez、F. Carrone、T. De Mattey、H. O. Demaestri 等，“Tutela：用于评估以太坊和 Tornado Cash 用户隐私的开源工具”，arXiv预印本 arXiv:2201.06811，2022年。[15] Z. Che、T. Li、M. Shen、H. Du 和 L. Zhu，“通过域不变特征学习关联以太坊混币服务中的账户”，arXiv预印本 arXiv:2505.09892，2025年。[16] X. Tu、T. Duan、S. Miao、H. Zhang 和 Y. Sun，“HiLoMix：用于混币地址关联的稳健高低频图学习框架”，arXiv预印本 arXiv:2511.07759，2025年。[17] K. Han、A. Xiao、E. Wu、J. Guo、C. Xu 和 Y. Wang，“Transformer中的Transformer”，《神经信息处理系统进展》，第 34 卷，第 15 908–15 919 页，2021年。[18] K. Cai、S. Lu 和 D. Qiu，“通过具有多种注意力机制的图神经网络进行动态图异构图上的链接预测”，载于《2022年第15届国际计算智能与设计研讨会（ISCID）论文集》。IEEE，2022年，第 62–65 页。[19] B. Perozzi、R. Al-Rfou 和 S. Skiena，“DeepWalk：社会表征的在线学习”，载于《第20届ACM SIGKDD知识发现与数据挖掘国际会议论文集》，2014年，第 701–710 页。[20] A. Grover 和 J. Leskovec，“Node2Vec：面向网络的可扩展特征学习”，载于《第22届ACM SIGKDD知识发现与数据挖掘国际会议论文集》，2016年，第 855–864 页。[21] T. N. Kipf 和 M. Welling，“使用图卷积网络进行半监督分类”，arXiv预印本 arXiv:1609.02907，2016年。[22] P. Velickovic、G. Cucurull、A. Casanova、A. Romero、P. Lio、Y. Bengio 等，“图注意力网络”，stat，第 1050 卷，第 48 550 页，2017年，第 20 期，第 10 页。[23] K. Xu、W. Hu、J. Leskovec 和 S. Jegelka，“图神经网络的能力有多强？”，arXiv预印本 arXiv:1810.00826，2018年。[24] W. Hamilton、Z. Ying 和 J. Leskovec，“大规模图上的归纳式表示学习”，《神经信息处理系统进展》，第 30 卷，2017年。[25] F. Manessi、A. Rozza 和 M. Manzo，“动态图卷积网络”，《模式识别》，第 97 卷，第 107000 页，2020年。

[26] R. Dey 和 F. M. Salem，“门控循环单元（GRU）神经网络的门变体”，载于 2017年IEEE第60届中西部电路与系统国际研讨会（MWSCAS）。IEEE，2017年，第 1597–1600 页。[27] A. Pareja、G. Domeniconi、J. Chen、T. Ma、T. Suzumura、H. Kanezashi、T. Kaler、T. Schardl 和 C. Leiserson，“EvolveGCN：面向动态图的演进式图卷积网络”，载于 AAAI人工智能会议论文集，第 34 卷，第 04 期，2020年，第 5363–5370 页。[28] P. Goyal、S. R. Chhetri 和 A. Canedo，“DynGraph2Vec：利用动态图表示学习捕获网络动态”，《知识型系统》，第 187 卷，文章编号 104816，2020年。[29] F. A. Gers、J. Schmidhuber 和 F. Cummins，“学会遗忘：使用长短期记忆网络进行持续预测”，《神经计算》，第 12 卷，第 10 期，第 2451–2471 页，2000年。[30] Y. Zhu、F. Cong、D. Zhang、W. Gong、Q. Lin、W. Feng、Y. Dong 和 J. Tang，“WinGNN：采用随机梯度聚合窗口的动态图神经网络”，载于第29届ACM SIGKDD知识发现与数据挖掘会议论文集，2023年，第 3650–3662 页。[31] D. Xu、C. Ruan、E. Korpeoglu、S. Kumar 和 K. Achan，“结合函数时间表示学习的自注意力”，《神经信息处理系统进展》，第 32 卷，2019年。[32] W. Cong、S. Zhang、J. Kang、B. Yuan、H. Wu、X. Zhou、H. Tong 和 M. Mahdavi，“时序网络真的需要复杂的模型架构吗？”，arXiv 预印本 arXiv:2302.11636，2023年。[33] D. Xu、C. Ruan、E. Korpeoglu、S. Kumar 和 K. Achan，“时序图上的归纳式表示学习”，arXiv 预印本 arXiv:2002.07962，2020年。[34] T. Hospedales、A. Antoniou、P. Micaelli 和 A. Storkey，“神经网络中的元学习：综述”，IEEE模式分析与机器智能汇刊，第 44 卷，第 9 期，第 5149–5169 页，2021年。[35] C. Finn、P. Abbeel 和 S. Levine，“用于深度网络快速适应的模型无关元学习”，载于国际机器学习会议。PMLR，2017年，第 1126–1135 页。[36] D. W. Hosmer Jr、S. Lemeshow 和 R. X. Sturdivant，《应用逻辑回归》。约翰·威立父子公司，2013年。[37] L. Breiman，“随机森林”，《机器学习》，第 45 卷，32，2001年，第 5 页。–。[38] J. L. Elman，“ 在时间中发现结构 ”，《认知科学》，第 14 卷，第 2 期，第 179–211 页，1990年。[39] J. R. Anderson，《人类心智如何存在于物理宇宙中？》。牛津大学出版社，2009年。[40] I. Higgins、L. Matthey、A. Pal、C. Burgess、X. Glorot、M. Botvinick、S. Mohamed 和 A. Lerchner，“β变分自编码器：通过受约束变分框架学习基本视觉概念”，载于国际学习表征会议，2017年。



Shuyi Miao 于2023年获得中国太原太原理工大学数据科学学院计算机科学与技术系理学硕士学位。目前，她正在中国北京的北京航空航天大学人工智能研究院攻读博士学位。她的研究兴趣包括区块链监管、图机器学习和知识图谱。



Wangjie Qiu 现任北京航空航天大学北京未来区块链与隐私计算高精尖创新中心副教授。他于2008年获得学士学位，并于2013年获得博士学位，两个学位均为北京航空航天大学数学专业。他的研究兴趣包括信息安全、区块链和隐私计算。他还担任中国通信学会区块链专业委员会副秘书长。作为创始团队成员，他成功研发了长安链，这是一种先进的国际区块链技术体系。他在信息安全、区块链和隐私计算领域拥有多项发明专利，并于2014年获得国家技术发明奖一等奖。



Xiaofan Tu 于2024年获得武汉理工大学学士学位。目前，他正在中国北京的北京航空航天大学人工智能研究院攻读理学硕士学位。其主要研究兴趣包括区块链、数据挖掘和图学习。



Yunze Li 于2023年获得北京工业大学学士学位。目前，他正在中国北京的北京航空航天大学人工智能研究院攻读理学硕士学位。其主要研究兴趣包括区块链、数据挖掘和图学习。



Yongxin Wen 于2023年获得中国太原太原理工大学信息与计算机学院理学硕士学位。他在中国沈阳东北大学获得通信工程学士学位。目前，他在中国北京的中国兵器工业第二〇八研究所担任工程师。其主要研究兴趣包括信道预测和反向散射通信。



Zhiming Zheng 于1987年获得中国北京北京大学数学科学学院数学博士学位。目前，他是中国北京航空航天大学人工智能研究院教授。他的研究兴趣包括精细智能、区块链和隐私计算。他是区块链-长安链的发起者之一，也是中国科学院院士。