

目录

论文 2 小结: (ToN-review) ReinShard: Efficient Structure-Aware Allocation for General Accounts in Sharded Blockchain via Deep Reinforcement Learning	1
(一) 阅读思想方面:	1
(二) 论文技术方面:	1
1. 知识学习:	2
2.1.1 分片思想到 shard blockchain	2
2.1.2 图神经网络 (GNN) 中的 GAT	2
2.1.3 深度强化学习 (RL) 中的 PPO	2
2.1.4 区块链系统设计: D-shard / W-shard 双层架构去中心化	2
2. 创新学习:	2
2.2.1 算法: 用“图注意力 + 强化学习”来减少跨片交易 (最关键创新)	2
2.2.2 系统: 把“AI 学习过程”做去中心化 (D-shard)	2
2.2.3 数据驱动的奖励函数: 来自真实 Ethereum 实证研究	3
(三) 论文行文方面	3
1. 总体行文思路应该是:	3
1.1 论文八股式:	3
1.2 核心部分 Sect4 行文:	3
2. Introduction	3
3. 实验图表 (突出)	4
3.2.1 流程图清晰易懂, 图形和颜色清晰	4
3.2.2 折线图配色区间, 突出 this work	4
3.2.3 热力图用得很好	4
3.2.4 箱线图是个啥	5
(四) 不足之处学习	6
1. 缺乏 adversarial (对抗性) 分析	6
2. 分片机制落地难度高	6
3. 模型复杂度和 scalability 分析不够	6

论文 2 小结: (ToN-review) ReinShard: Efficient Structure-Aware Allocation for General Accounts in Sharded Blockchain via Deep Reinforcement Learning

(一) 阅读思想方面:

1. 公式必须要知道参数表示和物理意义, 这样才能看出解决了什么问题, 推导过程可以暂时放一放。
2. 实验不用过分详细展开, 理解其图表表达的意思即可

(二) 论文技术方面:

1.知识学习：

2.1.1 分片思想到 shard blockchain

把交易系统拆成很多小区

为了提升性能，我们要使得：

- 1. 跨片交易减少 (**Cross-shard TX ↓**)：账户 A 和 B 常交互，就应该放在同一个 shard
- 2. 账户迁移减少 (**Migration ↓**)：不要让策略因为每次短时变化就让账户到处跑
- 3. 分片负载均衡 (**Load balance ↑**)：“热账户”要合理分散，避免集中在一个 shard

2.1.2 图神经网络 (GNN) 中的 GAT

它把“账户”当作图里的节点、把“交易关系”当作边，学会：“哪些账户关系更重要”，利用注意力权重找到隐含的交易依赖

GAT = 能看出哪两个人（账户）经常交易，所以应该放在一个分片里。

2.1.3 深度强化学习 (RL) 中的 PPO

- RL 是“让 AI 自己学”
- PPO 是 RL 中最稳定、收敛快的算法
- 在本论文中，PPO 用来决定“账户应该分到哪个分片”

2.1.4 区块链系统设计：D-shard / W-shard 双层架构去中心化

- D-shard = 专门训练 AI 的“决策小区” (Decider shard)
- W-shard = 负责正常交易处理 (Worker shard)
- D-shard 用 PBFT 达成一致的 AI 策略
- 所有 W-shard 都执行这个策略

区块链不只是执行区块，它本身内部也可以运行一个去中心化 AI 来做决策。

2.创新学习：

2.2.1 算法：用“图注意力 + 强化学习”来减少跨片交易（最关键创新）

- 用 GAT 提取交易依赖
- 用 PPO 学习如何更好地分配账户
- 目标是减少跨片、迁移、负载不均衡

这是之前工作没有做到的组合式方法。

2.2.2 系统：把“AI 学习过程”做去中心化 (D-shard)

- 不再需要一个中心 AI 服务器
- 训练过程由区块链节点共同完成
- 策略参数通过 PBFT 共识同步，使每个节点都有相同模型
- 攻击者无法污染学习过程

2.2.3 数据驱动的奖励函数：来自真实 Ethereum 实证研究

论文不是凭空设计 reward，而是实际统计了 14 类以太坊交易类型，给不同类型设定重要性权重，使用真实交易依赖图来训练 RL。

相比依赖仿真或随机模型，这是很大的创新点。

（三）论文行文方面

1. 总体行文思路应该是：

1.1 论文八股式：

Abstract
1. INTRODUCTION（引言）
2. RELATED WORK（相关研究：单独成块）
3. MEASUREMENTS ON REAL-WORLD ETHEREUM TXS: A FEASIBILITY STUDY（以太坊真实交易测量：方法可行性分析）（非常清晰）
4. REINSHARD DESIGN（系统设计：建模）
5. PROBLEM FORMULATION（问题形式化：数学分析）
6. ALGORITHM DESIGN（算法设计：数学分析）
7. SECURITY ANALYSIS（安全性理论分析）
8. EVALUATION（实验验证：全为性能测试）（写得很棒）
9. CONCLUSIONS（结论）
Reference

1.2 核心部分 Sect4 行文：

Part Two 02 研究设计

- 2.1 可行性研究
- 2.2 ReinShard 设计
- 2.3 数学模型与算法

2. Introduction

写得不错，层次分明，从这里挖掘可以讲得一清二楚。

3. 实验图表（突出）

很漂亮，对比明显，解释清晰：

3.2.1 流程图清晰易懂，图形和颜色清晰

优秀示范：

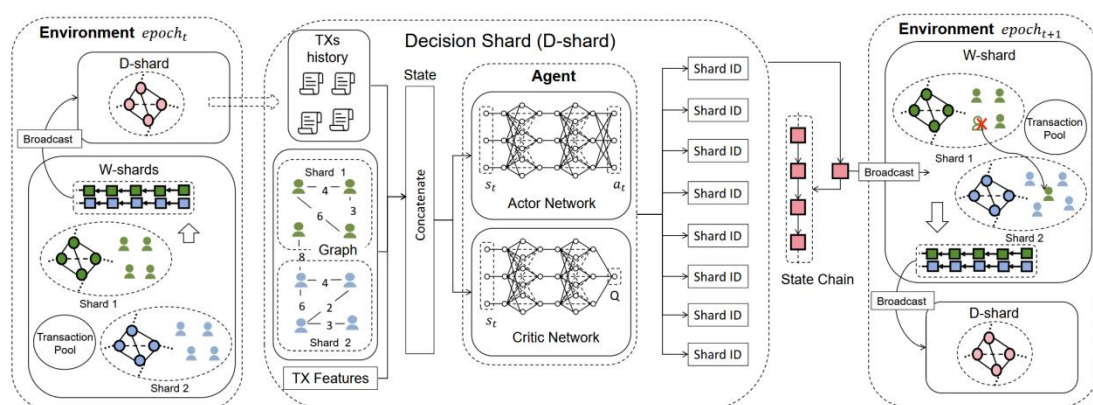


Fig. 4. Overview of the REINSHARD framework. Operating in epochs, it leverages a DRL agent to dynamically optimize account allocation strategies based on real-time transactional feedback.

3.2.2 折线图配色区间，突出 this work

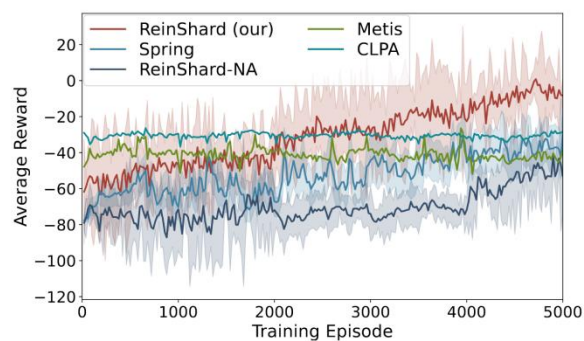


Fig. 6. Evaluation of training performance.

3.2.3 热力图用得很好

F. TX Distribution of Shards (分片的交易分布)

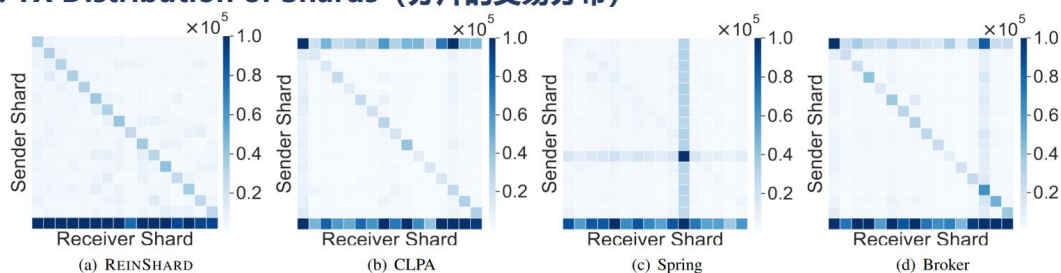


Fig. 11. The comparison of TX distribution under different account allocation strategies for 600,000 TXs.

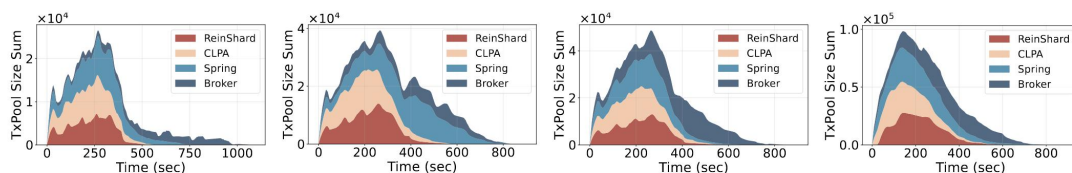
热力图 (heatmap) 展示了 600,000 TXs 在不同分片下的分布 (每个分片作为行或列, 矩阵主对角表示 intra-shard 交易) 以直观对比不同算法的聚类效果: 减少跨片交易与均衡负载方面的可视化

热力图的主对角线 (diagonal) 越深 → 表示更多的 intra-shard 交易 (好)。注: $Matrix[i][i]$ = shard i 内部的交易 (intra-shard) 代表 “本地交易”
行/列的总和 (Sum row) 越平滑/统一 → 表示 shard 负载均衡良好。注: $Sum[j]$ = shard j 的总系统负载 (所有 inbound/outbound) 深色代表热点 shard

实验结论

ReinShard 的热力图呈现更明显的主对角深色带, 说明其能把大量交互集中在同一 shard 内 (高 intra-shard clustering), 并且底部 “Sum” 行显示每个 shard 的总负载相对均匀, 从而同时兼顾少跨片与负载均衡。

相反, CLPA 与 Broker 的热力图显示更分散或存在部分 shard 过热的现象。



(a) TX injection rate = 3200 TXs/Sec (b) TX injection rate = 4000 TXs/Sec (c) TX injection rate = 5000 TXs/Sec (d) TX injection rate = 8000 TXs/Sec

Fig. 8. Queue size of TX pool while fixing $H = 16$ and varying TX injection rate within $\{3200, 4000, 5000, 8000\}$ TXs/Sec. The results of REINSHARD, CLPA, Spring, and Broker are (a) max = $\{7212.2, 9389.7, 9780.2, 2561.1\}$, avg = $\{1784.2, 2183.8, 2636.3, 1328.1\}$; (b) max = $\{13974.8, 15630.0, 13518.9, 6543.9\}$, avg = $\{3774.0, 3983.7, 5828.3, 2886.1\}$; (c) max = $\{13056.4, 14714.4, 14670.9, 13518.9\}$, avg = $\{3573.0, 3892.8, 4278.2, 5828.3\}$; and (d) max = $\{27817.7, 31172.3, 29877.5, 21569.8\}$, avg = $\{7858.6, 7485.2, 9655.7, 9579.3\}$, respectively.

3.2.4 箱线图是个啥

- **箱体**: 中间的矩形, 上下边分别是上四分位数 (Q3) 和下四分位数 (Q1) —— 箱体越窄, 说明数据越集中 (负载差异小)。
- **箱体里的线**: 中位数 —— 中位数越低, 说明数据的 “中间水平” 越好 (平均负载越低)。
- **须线 (箱体上下的竖线)**: 代表数据的 “合理范围” (通常是 $Q1 - 1.5 \times IQR$ 到 $Q3 + 1.5 \times IQR$, $IQR = Q3 - Q1$)。
- **须线外的点**: 异常值 —— 如果有很多异常值, 说明数据中有极端大 / 小的指标 (比如负载过高的分片)。

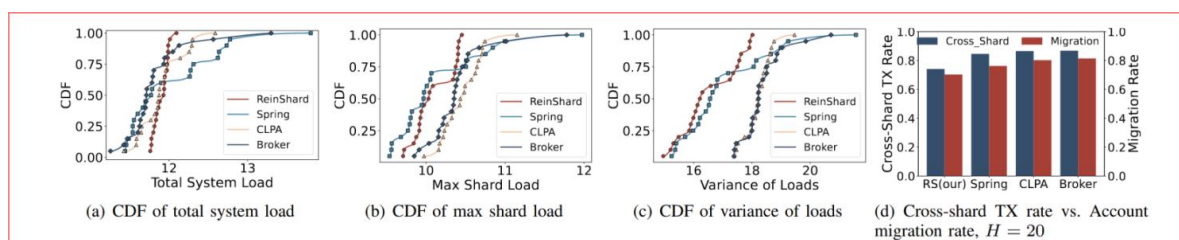


Fig. 9. Workload performance while fixing $H = 20$ and TX injection rate = 5000 TXs/Sec. In figures, CDF stands for cumulative distribution function.

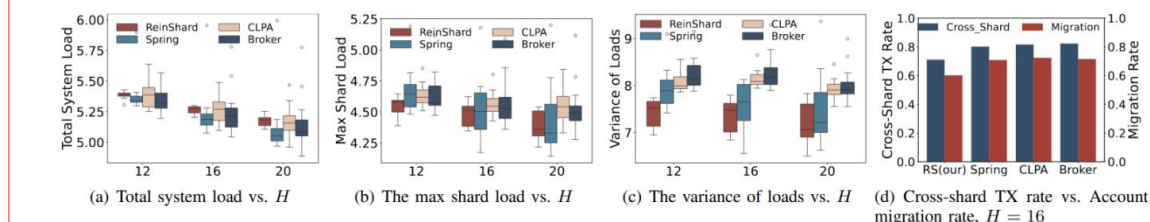


Fig. 10. Impact of shard number on total workload, maximum shard load, and load variance, while varying H within $\{12, 16, 20\}$, and fixing TX injection rate = 5000 TXs/Sec. Cross-shard TX rate and account migration rate are also evaluated when H is fixed at 16.

(四) 不足之处学习

优点在创新、行文和实验图表上，见上文

1. 缺乏 adversarial (对抗性) 分析

没考虑坏人可能会伪造交易关系、欺骗 GAT、污染 RL 训练。这是所有区块链论文必须讨论的点，缺失会被扣分。

2. 分片机制落地难度高

提出去中心化 RL 的架构很有创意，但实际部署 D-shard 训练成本可能很高；每轮都要进行 PBFT 共识，开销可观。真实区块链系统很难直接采用，需要进一步优化。

3. 模型复杂度和 scalability 分析不够

论文没有解释：GAT 在百万节点图上的复杂度、PPO 的收敛速度、随系统规模扩大训练成本如何变化、D-shard \rightarrow W-shard 参数同步的通信开销，对 ToN 级别论文来说，这些应该补充。