BLOCKS：基于区块链的支持跨孤岛知识共享以实现高效的 LLM 服务

Zhaojiacheng Zhou†, Hongze Liu†, Shijing Yuan‡, Hanning Zhang†, Jiong Lou†, Chentao Wu†, Jie Li§*
†Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
‡China Telecom Research Institute, Shanghai, China
§ MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University
Email: *{zzjc123, seniordriver233, 2019ysj, zhang5026871, lj1994, wuct, lijiecs}@sjtu.edu.cn,

摘要—大型语言模型（LLMs）的幻觉问题越来越受到关注。通过外部知识增强 LLMs 是解决这一问题的一种有前景的方法。然而，由于隐私和安全方面的担忧，大量的下游任务相关知识仍然分散在各种“孤岛”中，难以访问。为了弥合这一知识差距，我们提出了一种基于区块链的外部知识框架，该框架协调多个知识孤岛，为大型模型检索提供可靠的基础知识，并确保数据安全。技术上，我们将本地数据中的知识提炼成提示，并在区块链上执行交易和记录。此外，我们引入了信誉机制和交叉验证，以保证知识质量并激励参与。进一步地，我们设计了一个查询生成框架，为大型模型检索提供了直接的 API 接口。为了评估所提出的框架性能，我们在各种知识来源上进行了广泛的实验。结果表明，该框架在区块链环境中实现了高效的 LLM 服务知识共享。

***Index Terms*—LLM, 知识共享；区块链；声誉机制。**

I. 介绍

大型语言模型（LLMs）展示了令人印象深刻的人类般文本生成能力，但存在诸如幻觉——生成看似合理但实际上不正确的事实内容等显著限制 [1]。解决这一问题的一个有希望的方向是通过增强外部知识来改进 LLMs，这可以提高它们的准确性和可靠性。诸如检索增强生成（RAG）[2] 和知识链（CoK）[3] 等技术已被提出以有效地纳入此类知识。

大多数关于大语言模型增强的现有研究都集中在改进检索策略上，通常假设可以访问集中式、静态的数据集 [3] 。然而，新兴的研究表明，许多大语言模型应用需要多领域知识才能实现最佳性能 [3], [4] 。在实践中，与任务相关的知识常常分散在多个孤立的孤岛中，由于隐私问题、竞争利益和共享激励不足，使得访问变得困难。虽然有些系统可以从网络中提取有价值的信息 [5] ，但缺乏系统的跨孤岛知识共享机制极大地限制了它们的可扩展性和有效性。对于需要高价值领域特定知识的综合任务而言，这一问题尤为明显，这些知识大多未被充分利用。

统一孤立的知识以生成连贯且可操作的输入供 LLM 使用面临三个主要挑战：**P1: 激励机制**，**P2: 服务质量（QoS）**和 **P3: 安全性**。对于 **P1**，激励孤岛所有者贡献知识需要解决参与负担和隐私问题。对于 **P2**，将异构和非结构化数据集成到 LLM 兼容格式中必须同时保持数据实用性和隐私性。直接共享原始数据会带来隐私风险，并且由于冗余和结构性不一致可能会降低 LLM 性能，这也会导致推理延迟增加 [2]。此外，LLM 服务通常对延迟敏感，而知识共享容易受到网络动态带来的延迟影响。对于 **P3**，分布式贡献者的本质不信任性使得系统面临诸如提示注入攻击 [6] 等威胁。

区块链技术因其固有的透明性、不可变性和可追溯性，为解决这些挑战提供了有前景的基础 ~ [7]–[9]。通过利用共识协议和链上声誉机制，区块链可以提供去中心化的激励并确保参与者之间的安全交互 ~ [10]。然而，先前的努力——如协作数据库 ~ [11] 和分布式知识共享系统 ~ [12]——并未针对 LLM 特定的知识整合，并且由于高数据传输开销经常导致性能瓶颈。此外，大多数基于声誉的系统采用开环架构，这使得关键组件超出监控范围，从而引入了安全漏洞 ~ [13], [14]。

* Jie Li is the corresponding author.
The code is in: https://anonymous.4open.science/r/BLOCKS.

为此，我们提出一个基于 **BLO** 区块链的 **C** 跨孤岛 **K** 知识 **S** 分享框架 (**块**)，该框架建立在 COSMOS [15] 之上，旨在协调多个孤岛并为 LLM 服务提供可信、特定领域的知识。为了处理 **P1**（激励）和 **P3**（安全性），我们实现了一个基于智能合约的信誉机制，确保数据质量的同时激励参与。为进一步保障共享知识的完整性，我们引入了基于基准的验证过程，对低于预定义相似度阈值的贡献者进行惩罚。对于 **P2**（服务质量），BLOCKS 通过将本地数据转换为提示来促进安全的知识提炼，这些提示随后被上链交易并记录。此外，我们集成了一个缓存机制**缓存**，以提高频繁访问提示检索的效率。为了进一步优化区块链账本存储和并行请求处理，我们在 IAVL-Tree 之上实现了一个哈希桶键值存储层，存储由**缓存 PRO** 过滤和优先排序的知识。据我们所知，这是首个构建面向 LLMs 的基于区块链的外部知识框架的工作。我们的主要贡献总结如下：

- 我们提出了**块**，一个用于分布式知识共享的区块链平台，它集成了**缓存 PRO** 以提高提示检索效率。
- 我们引入了一种基于智能合约的信誉机制来确保安全并激励诚实参与。
- 我们在多种知识来源上评估了 BLOCKS，并展示了它在实现安全和高效的 LLM 增强方面的有效性。

论文组织结构： 第 II 节概述了我们工作的动机。第 III 节介绍了块的设计。第 IV 节详细说明了实现。第 V 节展示了实验结果。第 VI 节对论文进行了总结。

II. 动机

A. 多源知识用于 LLM 服务

整合外部知识对于提升大规模语言模型（LLMs）的性能至关重要，特别是在处理复杂下游任务和缓解诸如幻觉等问题时。越来越多的研究表明，结合多样化的知识来源显著增强了 LLMs 的能力 [2], [3], [16], [17]。特别是，最近的研究发现，让 LLMs 接触多源和跨学科的知识能大幅提高其在具有挑战性任务上的表现 [3]。

为了实证演示这一点，我们使用广泛采用的 BERTScore 度量标准 [18] 进行了一个初步实验，比较了在利用单源知识与多源知识时大语言模型的表现。评估是在来自真实问答 [19] 和维基问答 [20] 数据集的问题上进行的。结果如图 1 所示，多源知识提供始终优于单源配置。

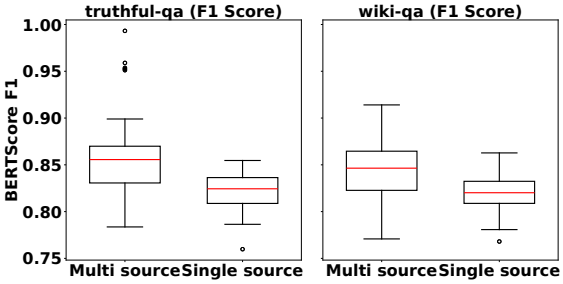

图 1: 通过多源知识整合提升大语言模型性能。

尽管其具有优势，多源知识的聚合引入了重大挑战，因为相关信息通常分散在不同的实体中 [11], [12] 。这突显了需要平台以实现对去中心化知识的有效、安全且保护隐私的访问。

B. 基于区块链的知识共享框架

在我们提出的框架中，区块链生态系统 [13], [14] 中的四个关键角色协作：

知识提供者：负责生成响应 LLM 服务请求提示的节点。

大型语言模型服务器：请求外部知识以满足用户查询的节点。

可信外部存储：归档历史知识并促进高效检索的组件。

验证者：验证提供提示的真实性和相关性，并评估提供商声誉的节点。

区块链的去中心化和不可变特性为可信且高效的知识共享提供了坚实的基础 [11], [12]。虽然先前的研究已经探索了跨领域知识交换的基于区块链的系统，但将此类机制整合到 LLM 服务中引入了几项特定领域的挑战，需要一个全面的设计来适应 LLM。

激励机制: 提供下游知识会带来计算和通信开销，因此实施公平奖励贡献者的激励机制至关重要。智能合约可以根据贡献的质量自动补偿，鼓励持续且有意义的参与 [10]。

服务质量（QoS）：确保及时响应 LLM 查询至关重要，特别是在可能影响数据所有者响应能力的网络状况波动的情况下。此外，将异构和非结构化的知识整合到 LLM 兼容格式中必须在保护用户隐私的同时保留数据效用。直接分享原始数据不仅存在隐私泄露的风险，还可能导致由于冗余和不一致而降低模型性能，最终增加推理延迟 [2]。

安全: 确保共享知识的完整性和真实性至关重要，特别是要防范恶意提示注入等威胁，如上下文混淆攻

击。现有框架中去中心化的贡献者本质上是不可信的，这增加了遭受此类攻击的风险 [6]。尽管之前的研究引入了声誉系统来解决这些问题 [13], [14]，但许多采用开放式设计，排除了持续监控的重要组成部分，从而未能填补安全漏洞。

III. BLOCKS 系统设计

在本节中，我们介绍了大型语言模型服务与单一原始数据所有者之间的交互，包括大型语言模型如何为外部知识请求生成查询以及数据所有者如何在不泄露隐私和暴露数据的情况下生成必要的知识。

A. 抵御提示攻击的声誉机制

随着 LLM 系统配备更多插件或访问模式，提示注入攻击的下游风险变得更高 [6]。为了利用分布式网络，我们可以实现交叉验证以检测提示注入攻击。为进一步提高系统的鲁棒性，我们引入了一个阈值检测机制，让验证者加速恶意节点信誉下降的过程。当验证者开始验证知识时，链会要求一个可信服务使用低成本模型评估知识的质量并生成一个阈值。如果结果超出该阈值，则直接将此节点识别为恶意节点并施加处罚。

由于区块链平台涉及多个具有不同角色的参与者，我们为每个角色设计了一个声誉系统。所有声誉值都存储在受信任的外部存储中，并且只能通过智能合约进行修改。

为了确保可靠的声誉评估，我们利用一致性指标（CS），基于诚实参与者所声称的期望值应保持一致这一假设。然而，由于个人能力的不同，报告值可能会有轻微的变化。一致性指标衡量验证者的评估是否与其他验证者对同一提示的评估相一致：

$$CS = \frac{\sum ||V - V_i|| \cdot R_{vi}}{(n-1) \cdot ||V_{\max} - V_{\min}|| \cdot R_{\max}} \tag{1}$$

其中，V 表示验证者分配的分数，而 V_i 则表示由验证者 i 给出的分数。R_{vi} 表示验证者 i 的声誉。$V_{\max}$, $V_{\min}$ 和 $R_{\max}$ 分别指验证者分配的最高分和最低分以及最高的验证者信誉。我们使用 m 和 n 来表示验证次数和大型语言模型服务的数量。

考虑验证数量的影响，我们引入置信度指标（CF）来量化风险作为标准差（STD）的 V：

为了确保知识的可信度，验证者检测提示攻击以保护知识质量不受恶意代理和低质量提供者的侵害。验证输出 V 在范围 $[0,1]$ 内进行量化。每个角色的信誉可以按如下方式计算：

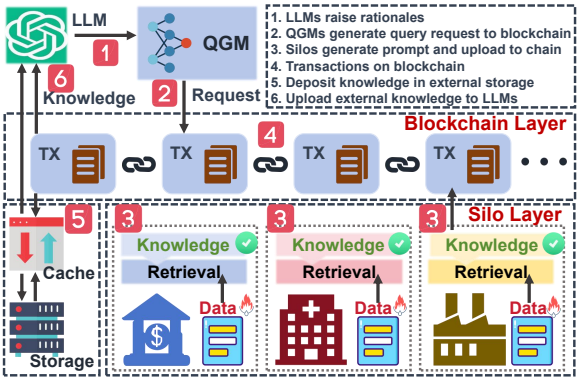

图 2: 系统概述

$$R_l = \alpha \cdot (1 - \text{Mean}(CS)) + (1-\alpha) \cdot R_l, \tag{2}$$

$$R_p = \frac{R_k + n \cdot \text{Mean}(V_i \cdot R_i) + m \cdot \text{Mean}(\text{Acc} \cdot R_l)}{n+m+1} - CF, \tag{3}$$

$$R_k = \alpha \cdot \text{Mean}(R_p) + (1-\alpha) \cdot R_k, \tag{4}$$

$$R_v = \alpha \cdot (1 - \text{Mean}(CS)) + (1-\alpha) \cdot R_v, \tag{5}$$

大型语言模型服务（R_l）、提示（R_p）、知识提供者（R_k）和验证者（R_v）的声誉是相互依赖并动态更新的。使用指数移动平均值，系数为 α，根据性能一致性、反馈和交叉验证计算声誉，定义如公式 2–5 所示。

为了确保知识的可信度、声誉的持久性，并为分布式知识验证提供足够的激励，我们引入了影响证明（PoI）共识机制。在此协议中，下一个区块由在知识质量上投入最高的知识提供者提出。奖励 C 按其影响力的比例分配给所有贡献的知识提供者，影响力定义为其声誉 R 与其相关提示的访问次数 A 的乘积：

$$C = R \cdot (\beta A_p + (1-\beta) A_v), \tag{6}$$

其中 $\beta \in [0,1]$ 是平衡提示生成 A_p 和验证 A_v 贡献的系数。

与工作量证明（PoW）经常因资源效率低下而受到批评不同，PoI 利用计算资源进行有意义的验证任务。它通过监控每个节点的影响来确保高效、目标驱动的共识过程，这种影响是根据其他节点访问其贡献提示的程度来衡量的。

B. 知识缓存设计

区块链平台在吞吐量方面固有限制，难以提供实时的外部服务给大语言模型。为了提高效率并实现及时响应，通常使用外部存储系统来补充区块链 [21]。传统的缓存管理策略如 LRU-k 和 LFU [22] 主要关注访问频率，但在对抗环境中可能失效。例如，恶意节点可能会反复访问低质量的提示，导致它们在缓存中持续存在，并可能放大其负面影响。

为解决此问题，我们提出了一种以**概率**iority**O**riented **缓存（PROCache）**为导向的策略，该策略将提示声誉、价值和成本整合到缓存优先级中。PROCache 包括两个主要组件：检索功能和缓存存储系统。检索功能使用 LlamaIndex [23] 实现，支持基于元数据和内容的提示嵌入、索引和检索。缓存存储包括历史队列、优先级队列和键值存储。PROCache 中最小的存储单位是一个缓存节点，由节点 ID、内容、元数据和优先级定义。

当缓存未命中时，仅将提示的哈希值、访问次数和轻量级历史记录存储在历史队列中以最小化空间使用。只有在历史队列中的访问次数达到 k 次后，才会将提示添加到缓存中。达到此阈值后，创建一个完整的缓存节点，包括提示内容、元数据、声誉和优先级。节点 ID 由提示的哈希值和计数导出，并以键值对的形式存储在主缓存中。

缓存在 PROCache 中通过优先队列进行管理，其中提示根据其计算的优先级分数排序。这种设计确保 PROCache 保留高质量、高价值的提示，同时减轻恶意活动引起的缓存污染。为了全面考虑访问频率、提示成本和存储大小，提示的优先级分数定义为：

$$\text{Priority} = \left(\frac{\text{Frequency} \times \text{Cost}}{\text{Size}} \right)^{R_t - R_b} \tag{7}$$

这里，R_b 是一个用于区分潜在恶意响应的声誉阈值，而 R_t 是提示当前的信誉分数。这种表述确保了提示在缓存中的推广不仅仅依赖于重复访问，而是通过质量和实用性的均衡评估来实现，使得系统更能抵御对抗性操纵。

C. 交易设计

一个完整的 BLOCKS 交易周期包含四个子交易，涉及用户、缓存、供应商和验证者。所有子交易的临时数据被封装在一个称为查询会话的统一结构中，由会话索引唯一标识。对于持久存储，区块链使用 COSMOS IAVL 树记录键值对，并利用不同的存储前缀来区分数据类型。实现定义了四种键值对类型：

数据表：存储用户提示及其 SHA-256 哈希，以及一个额外的哈希计数以区分碰撞。每个条目包括提示内容和其供应商。

声誉提示：使用与数据表相同的哈希和计数。每条记录包含提示的声誉和历史验证信息，其中每个验证包括验证者 ID、验证时验证者的声誉以及分配给该提示的分数。

供应商声誉 按供应商 ID 索引，该项目存储供应商当前的声誉得分。

声誉验证器 按验证者 ID 索引，这存储了验证者的声誉得分。

这种存储结构通过哈希索引高效地减少了大量数据冗余，确保使用短键快速检索，并通过额外的计数成功处理了哈希冲突。它保证了 COSMOS IAVL-Tree 存储中的紧凑性、性能和一致性。

IV. 把所有内容放在一起

A. BLOCKS 工作流

BLOCKS 框架遵循一个四阶段交易流程：创建会话，帖子缓存，更新知识和更新验证。

创建会话：用户提交一个带有其 ID、查询和支付的创建会话交易。LLM 使用查询生成模块（QGM）[3] 生成一个理由并记录在会话中。经过用户身份验证后，区块链广播一个新查询事件，并使用 COSMOS 银行模块扣除支付。

2. 帖子缓存：缓存服务监听新查询事件，通过缓存后交易以缓存内容和命中标志作出响应。在命中时，区块链存储内容并广播验证-更新事件。在未命中时，它广播知识更新事件请求知识供应商的输入。

3. 更新知识：供应商通过包含会话索引、供应商 ID 和检索内容的更新知识交易对知识 -更新 事件作出响应。在收到足够多的提交后，区块链通过更新验证事件启动验证。

4. 更新验证：验证者通过更新验证交易提交评估。官方验证者和普通验证者的反馈分别存储。一旦验证完成，区块链将最终确定会话，广播结果，更新声誉，并将会话数据移至持久化存储。

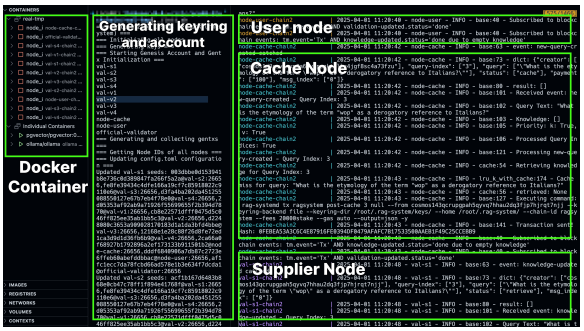

图 3: BLOCKS 的实现

图 2 描述了大语言模型、缓存服务、知识提供者和验证器之间的交互。经过验证的知识通过 PROCache 进行缓存，提高了响应效率。图 3 展示了我们的设备级实现。

对于缓存转售，外部存储在分配给提供者之前会收到服务费。奖励 U_p 的分配遵循提供者的声誉分数 R_p，由以下公式给出：

$$U_p = R_b \times \frac{R_p}{\sum R_p} \tag{8}$$

B. 安全性分析

我们考虑一个拜占庭式对手，该对手能够破坏系统中一部分节点，其中被破坏的节点数量由 $f < \frac{n}{3}$ 界定。在这一假设下，每个请求预期最多会从恶意实体接收到三分之一的响应。

本工作考察了几种对抗策略，其中一些受到了先前研究 [24] 的启发。主要的攻击向量概述如下：

自我推广： 一个恶意节点通过不诚实行为人为地提高自己的声誉分数，从而在系统中获得不应有的影响力。

勾结： 一组恶意节点合作以共同提升它们的声誉分数，从而增加其对系统的控制力。

诽谤： 一个被攻陷的领导节点试图降低诚实验证者的声誉分数，破坏系统中的信任和完整性。

假设 1. *恶意提示可以被识别，并且在由诚实的验证者评估时会收到比诚实提示更低的信誉分数。*

假设 2. *恶意验证者在验证过程中的优势并不超过诚实的验证者。形式上，声誉评估的预期偏差满足：*

$$E(|R_p^*, R_p^v|) \leq E(|R_p^*, R_p^m|) \tag{9}$$

其中 R_p^* 是提示 p 的真实信誉，R_p^v 是诚实验证者分配的信誉，而 R_p^m 是恶意验证者分配的信誉。

1) 声誉机制的安全性： 由于 PoI 机制直接依赖于信誉，我们首先分析信誉机制的安全性。

如方程 (2) 和方程 (5) 所示，提供失真反馈的恶意用户以及执行不正确验证的验证者随着时间推移声誉都会下降，最终在迭代 $t \to \infty$ 时达到零：

$$\lim_{t\to\infty} R_l^t = 0, \quad \lim_{t\to\infty} R_v^t = 0. \tag{10}$$

2) 影响证明的安全性： 我们分析 PoI 的安全性分为两个部分。

贡献计算： 知识提供者的总奖励 k 是：

$$W_k^t = C_k^t + \sum_{p\in P_k} A_p^t R_p^t. \tag{11}$$

由于 R_p^t 收敛到 V_p^*，恶意提供商经历声誉下降，因此得出：

$$\lim_{t\to\infty} W_{k_m}^t = 0, \quad \text{for all malicious providers.} \tag{12}$$

2. 声誉均衡的稳定性： 在无限的验证轮次中，假设诚实的验证者占主导地位，系统稳定为：

$$\forall p, \quad \lim_{t\to\infty} R_p^t = V_p^*. \tag{13}$$

因此，只有高信誉的提供者才能获得奖励，从而遏制恶意行为。

V. 实验

A. 实现

区块链基础设施。我们使用 COSMOS 实现我们的框架 [15], [25]，这使具有 Tendermint 共识的互操作区块链成为可能 [26]，支持治理、质押和 IBC。

数据集。维基 QA 的子集 [20]，TruthfulQA [19] 和 MathQA [27] 被使用。我们使用一个查询生成模型（QGM）来产生查询-答案对和嵌入，存储在一个 PostgreSQL 数据库中。

代理仿真。我们使用 Docker Compose 模拟所有区块链代理。交易通过 Tendermint RPC 进行通信。全节点包括用户、缓存和官方验证器节点。供应商和验证者作为 Tendermint 验证器运行。用户通过创建会话交易提交查询；缓存节点从存储中检索或转发给供应商。供应商节点使用 LlamaIndex [23] 提供提示。验证者对提示进行评分，而官方验证器应用余弦相似度来识别恶意行为。

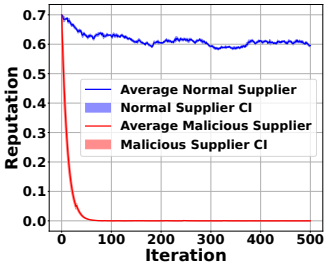

图 4: 供应商声誉

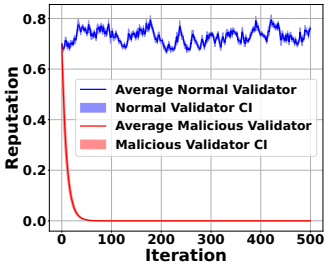

图 5: 验证者声誉

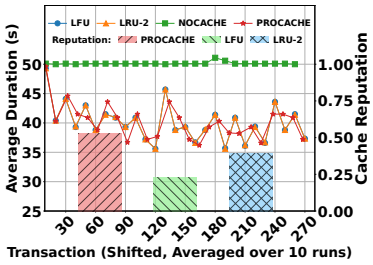

图 6: 高速缓存服务质量性能

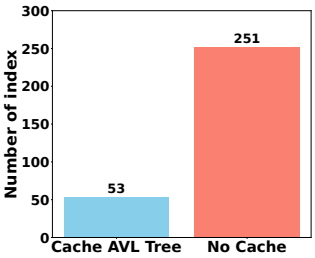

图 7: 存储改进。

环境： Ubuntu 24.04.1 LTS, Docker Compose v2.32.1, Python 3.10.16, Ollama v0.5.7, Ignite CLI v28.7.0, Cosmos SDK v0.50.11, CUDA 12.2 以及 RTX 4090 GPU。

B. 实验

声誉安全：我们通过分别测试供应商和验证者的性能来评估第 III-A 节提出的声誉机制。该设置包括八个诚实节点和三个恶意节点，它们执行在第 IV-B 节威胁模型中定义的对抗行为。诚实节点的声誉由平均值和置信区间 (CI) 曲线表示。结果如图 4 所示（针对知识提供者）以及图 5（针对验证者）。研究发现，恶意节点在所有对抗策略中的声誉均衰减至零，而诚实节点则保持稳定且显著更高的声誉，这验证了我们的声誉机制的有效性。

缓存性能：我们通过将提出的 PROCache 与基线 LFU 和 LRU-k 缓存策略进行比较来评估其性能。图 6 描述了缓存内容的性能指标和声誉，这些都是服务质量（QoS）密切相关的服务延迟和质量的指示器。PROCache 在减少延迟方面达到了可比性能，同时提高了缓存中的声誉。这表明我们的方案有效地使低质量的内容在引起广泛负面影响之前过时。

区块链存储：除了提高交易执行效率外，PROCache 还通过大约 80%的程度显著减轻了区块链账本的存储压力。我们通过对日志进行解析来评估缓存数据结构的修改，如图 7 所示。我们的分析显示，在 253 个不同的问题及其变体中，只有 53 个独特的提示索引以及它们的内容被存储在区块链上。这种减少归因于 PROCache 能够直接检索问题变体并重复使用供应商的历史答案，从而消除了存储重复的问题变体及其相应答案的需要。

VI. 结论

在这项工作中，我们提出了 BLOCKS，一个基于区块链的知识共享框架，该框架利用宇宙和 *Tendermint* 的能力，通过安全高效地整合外部多领域知识来增强大型语言模型（LLM）服务。通过解决诸如安全性、激励机制和服务质量（QoS）等关键挑战，BLOCKS 促进了分布式知识提供商之间的稳健合作。纳入基于声誉的验证机制确保了高质量知识的传递，营造了一个可靠和值得信赖的共享环境。为进一步提高服务质量，我们引入了一种新颖的知识缓存机制，该机制在保持存储效率和内容质量的同时实现了及时响应。

参考文献

[1] R. Patil and V. Gudivada, "A review of current trends, techniques, and challenges in large language models (llms)," *Appl. Sci.*, vol. 14, no. 5, p. 2074, 2024.

[2] S. Zhao, Y. Yang, Z. Wang, Z. He, L. K. Qiu, and L. Qiu, "Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely," *arXiv preprint arXiv:2409.14924*, 2024.

[3] X. Li, R. Zhao, Y. K. Chia, B. Ding, S. Joty, S. Poria, and L. Bing, "Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources," in *ICLR*, 2024.

[4] Z. Song, B. Yan, Y. Liu, M. Fang, M. Li, R. Yan, and X. Chen, "Injecting domain-specific knowledge into large language models: A comprehensive survey," *arXiv preprint arXiv:2502.10708*, 2025.

[5] H. Lai, X. Liu, I. L. Iong, S. Yao, Y. Chen, P. Shen, H. Yu, H. Zhang, X. Zhang, Y. Dong *et al.*, "Autowebglm: A large language model-based web navigating agent," in *Proc. 30th ACM SIGKDD*, 2024, pp. 5295–5306.

[6] J. Yi, Y. Xie, B. Zhu, K. Hines, E. Kiciman, G. Sun, X. Xie, and F. Wu, "Benchmarking and defending against indirect prompt injection attacks on large language models," *arXiv preprint arXiv:2312.14197*, 2023.

[7] S. Yuan, Q. Zhou, J. Li, S. Guo, H. Chen, C. Wu, and Y. Yang, "Adaptive incentive and resource allocation for blockchain-supported edge video streaming systems: A cooperative learning approach," *IEEE Trans. Mobile Comput.*, vol. 24, no. 2, pp. 539–556, Feb. 2025.

[8] S. Yuan, J. Li, and C. Wu, "Jora: Blockchain-based efficient joint computing offloading and resource allocation for edge video streaming systems," *Journal of Systems Architecture*, vol. 133, p. 102740, 2022.

[9] S. Yuan, J. Li, H. Chen, Z. Han, C. Wu, and Y. Zhang, "Jira: Joint incentive design and resource allocation for edge-based real-time video streaming systems," *IEEE Trans. Wireless Commun.*, vol. 22, no. 5, pp. 2901–2916, May 2023.

[10] T. L. Nguyen, L. Nguyen, T. Hoang, D. Bandara, Q. Wang, Q. Lu, X. Xu, L. Zhu, and S. Chen, "Blockchain-empowered trustworthy data sharing: Fundamentals, applications, and challenges," *ACM Comput. Surv.*, vol. 57, no. 8, pp. 1–36, 2025.

[11] W. Li, W. Tian, Z. Yan, Z. Li, J. Gao, F. Wu, J. Liu, W. Chen, and J. Ren, "Coraldb: A collaborative database for data sharing based on permissioned blockchain," *IEEE Trans. Mob. Comput.*, vol. 23, no. 9, pp. 8886–8901, 2024.

[12] G. Li, M. Dong, L. T. Yang, K. Ota, J. Wu, and J. Li, "Preserving edge knowledge sharing among iot services: A blockchain-based approach," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 4, no. 5, pp. 653–665, 2020.

[13] S. Khezr, A. Yassine, R. Benlamri, and M. S. Hossain, "An edge intelligent blockchain-based reputation system for iiot data ecosystem," *IEEE Trans. Ind. Inform.*, vol. 18, no. 11, pp. 8346–8355, 2022.

[14] W. Yang, C. Hou, Z. Zhang, X. Wang, and S. Chen, "Secure and efficient data sharing for iot based on blockchain and reputation mechanism," *IEEE Internet Things J.*, 2024.

[15] J. Kwon and E. Buchman, "Cosmos whitepaper," *A Netw. Distrib. Ledgers*, vol. 27, pp. 1–32, 2019.

[16] M. Wang, A. Stoll, L. Lange, H. Adel, H. Schütze, and J. Strötgen, "Bring your own knowledge: A survey of methods for llm knowledge expansion," *arXiv preprint arXiv:2502.12598*, 2025.

[17] L. Liu, J. Meng, and Y. Yang, "Llm technologies and information search," *J. Econ. Technol.*, vol. 2, pp. 269–277, 2024.

[18] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[19] S. Lin, J. Hilton, and O. Evans, "Truthfulqa: Measuring how models mimic human falsehoods," 2022. [Online]. Available: https://arxiv.org/abs/2109.07958

[20] Y. Yang, W.-t. Yih, and C. Meek, "Wikiqa: A challenge dataset for open-domain question answering," in *EMNLP*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 2013–2018.

[21] C. Xu, C. Zhang, J. Xu, and J. Pei, "Slimchain: scaling blockchain transactions through off-chain storage and parallel processing," *Proc. VLDB Endow.*, vol. 14, no. 11, p. 2314 – 2326, Jul. 2021.

[22] P. Panda, G. Patil, and B. Raveendran, "A survey on replacement strategies in cache memory for embedded systems," *2016 IEEE DISCOVER*, pp. 12–17, 2016.

[23] J. Liu, "LlamaIndex," 11 2022. [Online]. Available: https://github.com/jerryjliu/llama_index

[24] S. Hu, L. Hou, G. Chen, J. Weng, and J. Li, "Reputation-based distributed knowledge sharing system in blockchain," in *MobiQuitous*. New York, NY, USA: ACM, 2018, pp. 476–481.

[25] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," *Acm Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–41, 2021.

[26] E. Buchman, J. Kwon, and Z. Milosevic, "The latest gossip on bft consensus," 2019. [Online]. Available: https://arxiv.org/abs/1807.04938

[27] A. Amini, S. Gabriel, S. Lin, R. Koncel-Kedziorski, Y. Choi, and H. Hajishirzi, "Mathqa: Towards interpretable math word problem solving with operation-based formalisms," in *NAACL-HLT*. Association for Computational Linguistics, Jun. 2019, pp. 2357–2367. [Online]. Available: https://aclantholoy.org/N19-1245