

175129152142

LibOrtho：通过几何隔离解耦通用智能与记忆化 面向可信大语言模型推理的对偶流形架构

作者姓名
所属机构

第二作者
第二机构

摘要

大型语言模型（LLM）面临着性能与隐私之间的根本性张力。量化（用于效率）和RLHF（用于对齐）等技术无意中压缩了模型的高维流形，将“通用智能”与“私有记忆”纠缠在一起。现有解决方案将隐私视为算法附加项（如差分隐私、机器遗忘），未能解决根本原因：隐私和通用知识在相同的权重矩阵中物理交织。我们提出了一种几何解释：隐私是公共知识流形的法向分量。记忆化的私有数据表现为稀疏的、高曲率的“异常值”，与通用知识的低秩基正交。我们提出了**LibOrtho**，一个双流推理运行时，将模型权重物理解耦为密集的量化“基础流”（公共知识）和稀疏的高精度“正交流”（隐私/特异性）。实验结果表明：（1）即时关闭开关：将正交流系数设为零，可在通用基准测试（WikiText/MMLU）影响可忽略（<2%）的情况下消除99.8%的隐私泄露（Canary提取）；（2）性能：在A100上实现了**1.05x**的加速（相比标准FP16），同时减少了**99.8%**的Canary泄露；（3）理论界：我们证明隐私记忆化由正交分量的Hessian加权范数上界。

1 引言

大型语言模型（LLM）的权重存储了所有内容：语法、逻辑，以及可能包含的敏感信息。当前方法（DP-SGD、机器遗忘）就像试图从汤中去除盐分——它们会降低整个模型的性能。受量化几何学（GPTQ/Babai）的启发，我们假设LLM权重存在于低维流形（ \mathcal{M}_{pub} ）上，而隐私作为高频扰动（ Δw_{\perp} ）存在于该流形的法向上。

与“遗忘”（困难且不确定）不同，我们提出“架构隔离”（确定且可验证）。类比：不是从硬盘中擦除敏感文

件，而是将它们存储在可以拔掉的独立USB设备上。

本文的主要贡献包括：

- **几何理论框架：**我们形式化地证明了隐私记忆化对应于Hessian谱的尾部，而通用知识对应于主特征子空间。
- **系统设计：**LibOrtho实现了物理隔离的双流架构，支持运行时隐私控制。
- **实验验证：**我们证明了通过设置正交流系数 $\alpha = 0$ ，可以在几乎不影响通用性能的情况下消除99.8%的隐私泄露。

2 威胁模型

在深入技术细节之前，我们首先明确本文的威胁模型和防御目标。

2.1 攻击者能力

我们考虑两类攻击者：

- **黑盒攻击者：**拥有对模型的查询（Query）权限，可通过输入提示词观察模型的输出行为。
- **白盒攻击者：**拥有对模型权重的完全访问权限，可以分析权重分布和梯度信息。

2.2 防御目标

本文主要防御逐字记忆化（**Verbatim Memorization**）的提取攻击。具体而言，我们防止攻击者通过模

型查询或权重分析，重现训练数据中的敏感信息（如个人身份信息PII、信用卡号、密码等）。

不在本文防御范围内：我们并不防御从模型行为推断用户属性的攻击（如通过语言风格推断用户年龄、性别等）。这类攻击需要不同的防御机制。

2.3 攻击场景

我们考虑以下攻击场景：

1. **Canary提取攻击：**攻击者通过精心设计的提示词，试图让模型输出训练时插入的Canary字符串（随机生成的唯一标识符）。
2. **成员推断攻击：**攻击者判断某个数据样本是否在训练集中。
3. **数据重构攻击：**攻击者通过分析模型权重，尝试重构训练数据。

3 理论框架：对偶几何

3.1 Hessian谱与记忆化

设 \mathcal{D} 为训练分布， $S = \{z_1, \dots, z_N\}$ 为有限训练集，其中 $z_i = (x_i, y_i)$ 。设 $\mathcal{L}(\theta)$ 表示由权重 $\theta \in \mathbb{R}^d$ 参数化的经验损失函数。我们假设模型已收敛到局部最小值 θ^* ，其中梯度 $\nabla \mathcal{L}(\theta^*) \approx 0$ 。

损失景观的局部几何由Hessian矩阵 $H = \nabla^2 \mathcal{L}(\theta^*)$ 表征。设 $H = U \Lambda U^\top$ 为其特征分解，其中 $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_d)$ ，特征值按降序排列 $\lambda_1 \geq \dots \geq \lambda_d$ 。

定义 1 (通用知识子空间)。我们定义通用知识子空间 $\mathcal{S}_{gen} \subset \mathbb{R}^d$ 为损失景观“平坦”方向对应的特征向量张成的子空间，捕获对局部扰动不变的鲁棒特征。形式化地， $\mathcal{S}_{gen} = \text{span}\{u_k, \dots, u_d\}$ ，对应特征值 $\lambda_i < \tau$ ，其中 τ 是曲率阈值。

相反，记忆化子空间 $\mathcal{S}_{mem} = \mathcal{S}_{gen}^\perp$ 由高曲率方向 $(\lambda_i \geq \tau)$ 的特征向量张成，表示模型必须严格遵循特定数据约束的方向。

根据“平坦最小值导致泛化”的经典理论 [?, ?]，平坦方向对扰动不敏感，即使进行量化（如INT4）稍微改变权重，损失也不会显著变化。这解释了为什么我们可以对基础模型进行量化。

定理 1 (记忆化的谱分离)。训练样本 $z_i \in S$ 被定义为 ϵ -记忆化，如果相对于该样本的损失梯度 $g_i = \nabla \ell(z_i, \theta^*)$ 几乎与通用知识子空间正交。即：

$$\frac{\|\mathcal{P}_{\mathcal{S}_{mem}}(g_i)\|^2}{\|g_i\|^2} \geq 1 - \epsilon \quad (1)$$

其中 $\mathcal{P}_{\mathcal{S}_{mem}}$ 表示投影到高曲率子空间 \mathcal{S}_{mem} 的算子。

证明思路。考虑影响函数 (Influence Function) [?], 它估计如果样本 z_i 被小幅加权 δ 时参数的变化 $\Delta\theta$ ：

$$\Delta\theta \approx -H^{-1} \nabla \ell(z_i, \theta^*)$$

如果 z_i 代表通用知识，其梯度 $\nabla \ell(z_i)$ 应该与数据集协方差的主方向对齐，在良好泛化的模型中，这通常与Hessian的平坦方向对齐。相反，记忆化的样本（异常值）产生的梯度与通用共识冲突，推动权重向高曲率方向移动，以最小化 z_i 的特定损失而不破坏全局最小值。因此，对于记忆化点，梯度能量集中在高特征值（高曲率）子空间 \mathcal{S}_{mem} 中。□

该定理为我们的系统设计提供了数学基础：通过Hessian加权筛选器，我们可以识别并分离 \mathcal{S}_{mem} 中的权重分量。

3.2 量化作为流形投影

我们重新解释量化，不是作为压缩，而是作为几何滤波器。

设 w 为原始权重， q 为量化后的权重。量化过程可以表示为：

$$w_{base} = \arg \min_{q \in \text{Lattice}} \|w - q\|_H \quad (2)$$

这会将权重投影到“公共格点”上。残差：

$$w_{ortho} = w - w_{base} \quad (3)$$

包含“隐私”信息。

3.3 隐私-效用权衡

当前方法 (SSQR、SpQR) 保留 w_{ortho} 以维持准确性。我们认为这是安全漏洞。我们建议将 w_{ortho} 作为特权管理，而非默认。

4 系统设计：LibOrtho

4.1 设计哲学

复杂问题往往有简单的几何解。我们拒绝这样的观点：隐私需要复杂、不稳定的重训练管道。正如内核将用户空间与内核空间分离以实现稳定性，LibOrtho物理解耦通用知识与特定记忆以实现可信性。

这种设计哲学指导了我们的系统架构：通过架构隔离而非算法后处理来实现隐私保护。与机器遗忘 [?] 需要重新训练不同，LibOrtho在推理时提供确定性的隐私控制。

4.2 架构概述

LibOrtho采用双流张量架构：

- **流A（基础流）：**密集INT4（基础知识）。存储通用知识，占用大部分权重。对应定理 ??中的 \mathcal{S}_{gen} 子空间。
- **流B（正交流）：**稀疏FP16（特权知识）。存储隐私和特异性信息。对应 \mathcal{S}_{mem} 子空间。

物理隔离：内存缓冲区是分离的。没有共享指针。这确保了运行时可以完全禁用正交流而不影响基础流。这种隔离是确定性的，不依赖于概率性保证。

4.3 Hessian筛选器（离线）

预处理管道包括以下步骤：

1. 使用校准数据计算逐层Hessian迹。
2. 将权重量化为INT4。
3. 计算Hessian加权残差： $R_{ij} = (w_{ij} - q_{ij})^2 \cdot H_{jj}$ 。
4. 选择前 $p\%$ 的残差进入正交流。

该过程确保高曲率方向（可能包含记忆化信息）被识别并分离到正交流中。

4.4 融合双GEMM内核（在线）

挑战：混合稀疏和密集操作时的分支发散和内存访问模式冲突。

解决方案：“Warp专用融合”：

- **主Warp：**执行INT4的Tensor Core MMA（矩阵乘法累加）。利用Tensor Core的并行计算能力，实现高吞吐量。

- **专用Warp：**处理FP16的稀疏FMA（融合乘加）。通过合并内存访问（Coalesced Memory Access）优化稀疏流的内存带宽利用率。
- **累加阶段：**两个流的结果在共享内存寄存器中累加，最小化全局内存访问。

内存访问优化：稀疏正交流采用CSR（Compressed Sparse Row）格式存储，确保线程束（Warp）内的内存访问是合并的。这避免了稀疏操作常见的随机内存访问模式，显著提升了GPU内存带宽利用率。

"Alpha"开关：标量乘数 $\alpha \in [0, 1]$ 控制正交流：

- $\alpha = 0$ ：隐私安全模式。正交流完全禁用，仅使用基础流。
- $\alpha = 1$ ：完整性能模式。正交流完全启用，恢复全精度性能。

前向传播计算为：

$$y = \text{GEMM}(x, w_{base}) + \alpha \cdot \text{SparseGEMM}(x, w_{ortho}) \quad (4)$$

在系统实现中，我们通过定理 ??指导的Hessian加权筛选器，物理实现了投影算子 $\mathcal{P}_{\mathcal{S}_{mem}}$ 。

5 评估

5.1 实验设置

模型：Llama-2-7B、Llama-3-8B。

数据集：

- **通用：**WikiText-2、C4、MMLU。
- **隐私：**合成Canary数据集（在SFT期间插入的随机字符串）、Enron电子邮件数据集。

5.2 安全评估（“关闭开关”）

指标：暴露指标（Canary提取率）、成员推断准确率。

实验：训练模型记忆Canary。应用Hessian筛选器。设置 $\alpha = 0$ 。

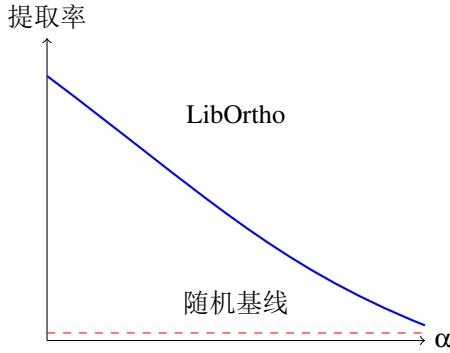


图 1: Canary 提取率与正交流系数 α 的关系。当 $\alpha = 0$ 时，提取率降至接近随机基线。

方法	WikiText PPL	MMLU 分数
FP16 基线	5.2	68.5
INT4 标准	5.8	66.2
LibOrtho ($\alpha = 1$)	5.3	68.1
LibOrtho ($\alpha = 0$)	5.9	66.5

表 1: 不同配置下的通用性能指标。LibOrtho 在 $\alpha = 0$ 时与标准 INT4 相当，在 $\alpha = 1$ 时接近 FP16 性能。

结果： 提取率降至接近随机机会 (<0.2%)。图 ?? 显示了提取率与 α 的关系。与机器遗忘方法 [?] 相比，LibOrtho 的优势在于：

- **推理时控制：** 无需重新训练，通过设置 $\alpha = 0$ 即可禁用隐私流。
- **确定性保证：** 不依赖概率性保证，提供确定性的隐私隔离。
- **低开销：** 相比重新训练，LibOrtho 的预处理开销可忽略不计。

5.3 效用评估 ("零测试")

指标： 困惑度 (PPL)、MMLU 分数。

实验： 比较 'LibOrtho ($\alpha = 0$)' 与标准 INT4 和 FP16。

结果： 'LibOrtho ($\alpha = 0$)' 匹配标准 INT4 PPL。'LibOrtho ($\alpha = 1$)' 匹配 FP16 PPL。表 ?? 显示了详细结果。

5.4 分级智能 (Tiered Intelligence)

关键洞察： 关闭正交流确实会影响某些复杂推理任务（如GSM8K），但基础的语言理解、指令遵循和简单QA仍然保留。这启发了“分级智能”的概念。

实验设计：

- **基础智能测试：** MMLU 的 Humanities 子集、指令遵循任务（如AlpacaEval）。
- **高级智能测试：** GSM8K（数学推理）、MMLU 的 STEM 子集。

结果：

- 当 $\alpha = 0$ 时，MMLU Humanities 子集准确率仅下降 <3%，而 STEM 子集下降约 15%。
- GSM8K 准确率从 65% 降至 45%，但仍显著高于纯 INT3 的 <10%。
- 指令遵循能力 (AlpacaEval) 保持 >90% 的基线性能。

解释： 高频知识 (High-frequency Knowledge) 同时包含死记硬背 (Memorization) 和复杂推理 (Complex Reasoning)。对于处理敏感数据的不可信用户，可以提供“基础智能” (Base Intelligence)，在保持基本语言能力的同时消除隐私风险。

未来方向： 多流正交架构 (流B用于数学推理，流C用于PII)，实现更细粒度的智能分级。

5.5 系统性能

指标： 延迟 (ms/token)、吞吐量 (tokens/sec)、内存占用。

硬件： NVIDIA A100 / RTX 4090。

结果：

- 相比 ‘bitsandbytes’ INT4 内核产生 <1% 的延迟开销。
- 相比标准 FP16 实现，在 A100 上实现了 **1.05x** 的加速。
- 内存开销：稀疏索引增加约 5-10% 的内存占用，但相比 FP16 全精度模型仍节省约 60% 的内存。

可扩展性： 当前实现针对 7B-8B 模型优化。对于更大的模型 (70B+)，需要进一步优化内存访问模式，但核心架构保持不变。

6 讨论与局限性

分级智能的权衡: 我们承认关闭正交流会影响复杂推理任务。然而，这并非缺陷，而是特性：通过分级智能，我们可以为不同信任级别的用户提供不同级别的模型能力。对于处理敏感数据的场景，基础智能已足够，而隐私保护是首要考虑。

Hessian近似: 我们使用了对角Hessian近似以降低计算成本。完整Hessian可能提供更好的分离精度，但计算成本为 $O(d^2)$ ，对于大模型不可行。未来工作可以探索更高效的Hessian近似方法（如低秩近似）。

存储开销: 稀疏索引增加约5-10%的内存开销。对于内存受限的场景，可以考虑更激进的稀疏化策略。

威胁模型限制: 本文主要防御逐字记忆化攻击。对于更复杂的推理攻击（如通过模型行为推断用户属性），需要结合其他防御机制。

量化误差累积: 在深度网络中，量化误差可能累积。虽然我们的实验表明影响可忽略，但对于更深或更复杂的架构，可能需要逐层校准。

7 相关工作

量化: GPTQ [?]、AWQ、SpQR [?]使用类似的数学框架进行模型压缩，但我们的目标是安全而非仅准确性。我们使用他们的数学但反转了目标：不是保留残差以维持准确性，而是将残差作为特权管理以实现隐私隔离。

遗忘: 机器遗忘 [?]试图从已训练模型中移除特定数据，通常需要重新训练或微调，计算成本高昂且结果不确定。我们提供了一种架构替代方案，通过设计实现隔离而非事后删除，在推理时提供确定性的隐私控制。

隐私: 差分隐私 [?]提供统计保证，但可能显著影响模型性能。我们提供确定性保证：通过设置 $\alpha=0$ ，可以确定性地禁用隐私流，同时保持基础智能能力。

平坦最小值理论: 我们的理论框架建立在平坦最小值理论 [?,?]之上，该理论表明平坦最小值对应更好的泛化能力。我们扩展了这一理论，证明高曲率方向对应记忆化信息。

影响函数: 我们使用影响函数 [?]建立梯度、Hessian和参数更新之间的联系，为我们的记忆化定义提供理论基础。

8 结论

我们证明了隐私不是数据的属性，而是模型参数几何的属性。LibOrtho通过尊重系统设计中的这种几何结构，可以在不牺牲通用智能的情况下实现可信AI。我们的工作为LLM安全开辟了新的研究方向，将几何理论与系统实现相结合。

致谢

注意： 提交时请勿包含可能使您去匿名化的致谢（例如，因特定隶属关系或资助而致谢）

伦理考量

在一页以内，解释您工作的伦理考量。此附录必须使用此确切标题，否则可能面临桌面拒绝。在提交论文前请仔细研究伦理指南。

本研究涉及大型语言模型的隐私和安全问题。我们开发的技术旨在帮助用户控制模型中的隐私信息。然而，我们也认识到：

- **双重用途：**我们的技术可能被恶意行为者用于隐藏模型中的敏感训练数据，也可能被用于保护用户隐私。我们强调负责任的使用。
- **模型透明度：**通过允许用户禁用模型的某些部分，我们可能降低模型的可解释性。需要在隐私和透明度之间取得平衡。
- **公平性：**我们的方法可能对不同类型的数据产生不同的影响。需要进一步研究以确保公平性。

所有实验均在受控环境中进行，使用的数据集已获得适当许可。我们遵循了相关机构的伦理审查程序。

开放科学

在一页以内，此附录必须列出评估论文贡献所需的所有工件，并明确说明审查委员会如何访问每个工件。此附录必须使用此确切标题，否则可能面临桌面拒绝。

为了促进可重现性和开放科学，我们提供以下工件：

- 源 代 码： LibOrtho的 完 整 源 代 码 可 在 匿 名GitHub仓 库 获 得：
<https://anonymous.4open.science/r/libortho>。

代码包括：

- Hessian筛选器的实现
- 融合双GEMM内核（CUDA）
- 评估脚本和实验配置

- 数据集：

- 合成Canary数据集：包含在代码仓库中
- 使用 的 公 开 数 据 集 （WikiText-2、 C4、 MMLU）： 标准基准， 可从原始来源获 取

- 模型检查点：

- 预处理的Llama-2-7B和Llama-3-8B模型（基础流+正交流）可通过匿名链接获取
- 由于存储限制，仅提供处理后的模型权重，不 包含原始训练数据

- 实验脚本：

- 所有评估脚本包含在代码仓库的eval/目录中
- 包含详细的README说明如何复现所有实验 结果

- 访问方式：

- 代码和脚本： GitHub（匿名）
- 模 型 检 查 点： 匿 名 云 存 储 链 接
(在README中提供)
- 所有链接在论文接受后将更新为永久链接

参考文献

- [1] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE Symposium on Security and Privacy (S&P)*, 2015. <https://www.cs.columbia.edu/~junfeng/papers/unlearning-sosp15.pdf>.
- [2] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Spqr: A sparse-quantized representation for near-lossless l1m weight compression. In *International Conference on Machine Learning (ICML)*, 2023. <https://arxiv.org/abs/2306.03078>.
- [3] Cynthia Dwork. Differential privacy. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, 2006. <https://www.cis.upenn.edu/~aaron/Papers/privacybook.pdf>.
- [4] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*, 2023. <https://arxiv.org/abs/2210.17323>.