

# 萧尧

(+86) 186-2182-3612 | ✉ [yaoxiao@g.harvard.edu](mailto:yaoxiao@g.harvard.edu) | 🏠 [charlie-xiao.github.io](https://github.com/charlie-xiao) | 🗣️ Charlie-XIAO | 📄 [yao-xiao-200073244](https://yao-xiao-200073244.github.io/)

## 教育背景

哈佛大学 | 理学硕士 | 计算科学与工程

2024.09 – 2026.05 (预期)

- GPA: 3.92/4.00, 相关课程包括: 计算机网络、数据系统、高性能计算、分布式系统 (MIT) 等。

纽约大学 | 理学学士 | 荣誉数学、计算机科学

上海 | 纽约 | 2020.09 – 2024.05

- 荣誉数学 GPA: 4.00/4.00, 相关课程包括: 线性代数、抽象代数、数学建模、概率论、数值分析、实/复分析、随机分析等。
- 计算机科学 GPA: 3.97/4.00, 相关课程包括: 数据结构、算法、随机化算法、计算机架构、操作系统、机器学习、软件工程等。

## 技术能力

- [1] 编程语言: Python, Rust, JavaScript/TypeScript, C, C++, Go; SQL, Java, MATLAB, Julia
- [2] 框架与库: Tauri, React; Numpy, Pandas, Polars, Scikit-learn, PyTorch; CUDA; SIMD/AVX; OpenMP, MPI
- [3] DevOps: Docker; Git; AWS, Google Cloud; Ansible; Kubernetes; GitHub Actions, CI/CD; Linux

## 工作经历

Scikit-learn | 开源项目 | 核心开发者 | 128 个已合并 PR

2023.04 – 至今

技能: Python, Cython, JavaScript, Sphinx, scikit-learn, numpy, scipy, pandas, polars, CI/CD

- 负责维护任务, 如测试套件覆盖率、代码重构、开发者 API 改进、自动化 GitHub 工作流等。
- 增强对稀疏数组和 polars 的支持、改进估计器的表达方式、优化指标函数的可视化、支持多标签数据交叉验证等。
- 优化增量主成分分析在稀疏数据上的性能 (速度 10x, 内存占用 3%); 改进半正定矩阵生成器 (内存占用 10%) 等。
- 对 scikit-learn 官方文档和网站的进行了整体改版, 并负责协调后续文档内容和 UI/UX 的改进工作。

复旦大学 DISC 实验室 | 实验室助理 | DASFAA'24 | GitHub

2023.05 – 2023.08

技能: Python, PyTorch, HuggingFace, 大语言模型 (LLM), 指令微调, 检索增强

- 构建 40 万条法律知识指令数据, 并采用法律三段论提示等技巧以提高模型回复的专业性。
- 微调 DISC-LawLLM (基于 Baichuan 13B Chat 的司法领域大模型), 性能超越 GPT-3.5 Turbo (当时最先进的通用模型)。
- 参与设计可验证知识检索模块, 引入外部知识库进行检索增强, 提高模型输出的真实性并减轻其幻觉。
- 推动了司法领域大模型系统评测基准的实现, 全面覆盖了多个客观和主观的评测维度。

## 科研经历

通过匿名化实现隐私保护的网路配置共享 | SIGCOMM'24 | GitHub

2022.10 – 2024.08

指导教授: 刘古月教授, [guyue.liu@gmail.com](mailto:guyue.liu@gmail.com)

- 提出并实现了 ConfMask 框架, 系统性地对网络配置中的拓扑与路由信息进行匿名化处理。
- 设计适用于不同网络路由协议 (OSPF, BGP, EIGRP 等) 的网络配置匿名化算法, 在降低去匿名化风险的同时, 保持关键网络功能。
- 严格证明 ConfMask 能够保证等效的网络路由, 并维持可达性、多路径一致性、路径长度等性质以确保共享的网络配置的可用性。

## 项目经历

基于 Raft 共识算法的高容错键值存储系统 | 课程项目

2025.02 – 至今

技能: Go, RPC, 分布式系统, 共识算法, 系统容错

- 使用 Go 语言开发了一个分布式键值存储系统, 并通过 Raft 共识算法确保系统的强一致性。
- 实现了领导节点选举、日志复制和状态机更新等机制, 可在节点故障和网络分区等情况下继续正常运行。
- 利用 goroutines 和 channels 实现并发高效的 I/O 处理、RPC 通信及容错机制。
- 在 MIT 6.5840 提供的测试框架下验证该系统设计, 确保其在多种故障场景下具备正确性、可靠性与高性能。

列存关系型数据库管理系统 | 课程项目 | GitHub

2024.09 – 至今

技能: C, C++, SIMD/AVX, OpenMP, MPI, 数据库分片, 缓存感知算法

- 使用 OpenMP 并行化与 SIMD 向量化复杂 SELECT 语句的执行, 在 1 亿条数据与 100 条筛选条件下提速 >20 倍。
- 支持 B+ 树索引, 在 1 亿条数据下批量加载开销 <20 毫秒, 对 5% 选择率的 SELECT 语句执行提速 >25 倍。
- 高度并行化 Radix Hash Join 算法, 在对 1 亿规模的两表执行 JOIN 语句时候, 相比传统哈希联接提速 >15 倍。
- 基于 MPI 进行数据库分片并实现多节点分布式处理, 达到近线性的提速效果和数据规模可扩展性。

Deskulpt: 跨平台的桌面定制工具 | GitHub

2024.03 – 至今

技能: Rust, TypeScript, Tauri, React, Vite, 组件打包工具, 插件系统 | 全栈开发

- 主导开发 Deskulpt, 一款基于 Tauri 的跨平台桌面定制工具, 支持用户使用 React 编写桌面小组件。
- 利用 IPC 和自定义的通信协议, 设计了一套插件系统, 使后端同时保持轻量化和高度可扩展性。
- 使用 Rust 构建基于 Rolldown 的桌面组件打包工具, 支持组件热重载、外部依赖管理、组件间共享 React 运行时等。
- 利用 Rust 异步确保 UI 响应流畅, 实现桌面组件打包、渲染等多任务的高效并发执行。
- 在 Deskulpt 内集成丰富的开发工具, 支持桌面组件和插件的创建、检索、调试、打包和分发。