

高康嘉

2027 届本科 | 预计推免 | 专业排名: 1/84

19033190031 | kkkzbh@gmail.com | github.com/kkkzbh



教育背景

2023.09-至今 河北大学 计算机科学与技术 (本科)

专业成绩: GPA 4.51/5 (专业第1/84)

主修课程: 程序设计 (92), 数据结构 (96), 计算机组成与体系结构 (88), 电子线路与数字逻辑 (98), 计算机通信与网络 (91), 离散数学 (96), 微机原理与汇编语言 (91), 操作系统 (94), 编译原理 (98), 高等数学 (97), 线性代数 (99), 概率统计 (96)

英语: 六级 473 分; CCF-CSP 认证: 350 分 (全国排名 1.88%)

荣誉奖项

• 国家奖学金		2025 年 12 月
• ICPC 国际大学生程序设计竞赛全国邀请赛	铜奖	2026 年 05 月
• CCPC 中国大学生程序设计竞赛全国邀请赛	银奖	2026 年 05 月
• CCPC 中国大学生程序设计竞赛河北省赛	金奖	2026 年 05 月
• 百度之星程序设计大赛决赛 (人才专项赛道)	国赛银奖	2025 年 08 月
• 中国高校计算机大赛-团体程序设计天梯赛	个人国二等奖 / 团队国二等奖	2026 年 05 月
• 码蹄杯程序设计大赛 (本科院校赛道)	国赛银奖	2025 年 08 月
• 蓝桥杯全国软件和信息技术专业人才大赛 (C/C++ 大学 A 组)	省一等奖	2025 年 05 月
• 全国大学生数学竞赛 (非数学 A 类)	省一等奖	2024 年 12 月
• 全国大学生数学建模竞赛	省二等奖	2025 年 09 月

科研经历

- Frequency-Structured Response Consolidation for Lifelong Person Re-Identification**
[河北大学智能感知与信息处理实验室](#) AAAI 2027 在投(CCF-A 会议)、第一作者 2025.09-至今
背景: 行人重识别旨在跨摄像头检索同一行人, 而终身行人重识别面临的最大问题是灾难性遗忘, 要求模型持续学习新域时保持在早期域上的性能。现有 LReID 方法为了规避隐私问题, 在学习新域时, 将旧域图像风格迁移至新域图像来共同参与训练, 进一步缓解灾难性遗忘。近期频域研究表明, 域风格会对不同频率成分产生结构化影响, 且各频段的跨域迁移能力不同。
动机: 我观察到新域图像与旧域风格图像之间的身份表征相同, 同时携带了一种尚未被充分利用的跨域变化信息, 提出 FSRC 框架。该框架向模型中嵌入一个结构化频率调制模块, 选择性地调节由风格变化引起的特征响应变化, 使模型能适应持续变化的域风格。同时提出跨风格对齐方法, 对同一行人在当前风格与历史风格下的成对特征响应进行对齐, 约束后续表征更新, 减少对已有知识的干扰从而支持知识巩固效果。目前在标准 LReID 基准上取得最优平均性能, 较现有最优方法的已见域/未见域 mAP 最高提升 1.2/0.9 个百分点。
总结: 在这段科研经历中, 我主要围绕终身行人重识别、灾难性遗忘、知识蒸馏、域泛化、跨模态 ReID 等领域系统阅读了约 50 篇相关论文, 成功复现过 AKA、PatchKD、DASK、LSTCK、ADSR 等 LReID 框架, 也经历了大量的失败实验与踩坑, 但这让我系统地明白了科研流程, 积累了一些科研经验, 在未来也会帮助我避开那些失败与踩坑的经历。
个人贡献: 全程由我在导师指导下独立完成, 包括阅读文献、发现问题、确定创新点、大量开展实验、论文撰写与图表制作。
- Spatial-Identity Supervision Calibration for Weakly Supervised Person Search**
[河北大学智能感知与信息处理实验室](#) MMAsia 2026 在投(CCF-C 会议)、第三作者 2026.02-至今
介绍: 围绕弱监督行人搜索任务, 关注标注不足条件下空间定位监督与身份判别监督不充分、不一致的问题, 尝试通过空间信息与身份信息的监督校准提升行人定位和身份匹配能力。
个人贡献: 参与部分消融与实验、部分论文图表制作

项目经历

- 面向编程教育的个性化学习 Agent 系统 (省级大创)** 项目负责人 [GitHub](#) 2024.11-2026.05
介绍: 项目指导教师为数据结构老师, 面向数据结构课程教学场景设计。分析学生的考试测试信息数据, 构建学生个人的能力画像, 学生通过与 Agent 交互获取学习报告、习题推荐、学习路径规划、薄弱点分析等分析信息, 也可与 Agent 交互进一步完善数据结构的学习。目前已尝试投入一次实际教学使用, 收集了教师与学生反馈, 对教师掌握学生学习情况, 以及对学生学习兴趣、易错薄弱点改进均有一定效果。
总结: 这个项目与 Codex、Claude Code 等编码智能体的区别在于, 后者基于聊天来编写代码, 而我们结合课堂教学数据, 试图探索一种区别于传统对话聊天且方便用于学习的 Agent 交互方式。
个人贡献: 项目方案设计、前端界面、数据导入与分析、能力画像构建、Agent 框架搭建与交互逻辑设计

- **x86/i386 操作系统** 独立完成 [GitHub](#) 2025.07-2025.11
介绍: 出于对操作系统原理的浓厚兴趣,使用 NASM 汇编、C 与现代 C++ 从零实现了一个 i386 操作系统。包含 MBR/Loader 引导、内核初始化、中断管理、分页内存、线程/进程调度、系统调用、IDE 磁盘驱动、文件系统和用户程序执行等。该操作系统可用 QEMU 执行,支持 ls/cd/pwd/ps/clear/mkdir/rmdir/rm/cat 等 Linux 常用命令,并支持通过 fork/exec/wait 来完成用户态程序的加载、执行与回收。操作系统的完整设计可见: [os.kkkzbh.cn](#)。
总结: 这个项目让我系统理解了操作系统的基本运行原理,并形成了从底层机制出发分析和设计程序的系统思维。
- **智能 QQ AI 代理** 独立完成 [GitHub](#) 2026.02-至今
介绍: 该项目以 Koishi、ChatLuna 为基础,实现了一个智能 QQ AI 代理。自主设计对话协议,支持语音/图片/文件等多模态消息输入与输出。支持 MCP、Skills 和插件功能,同时自主设计记忆系统与搜索功能。
另外我将学校的教务系统、学习通、志愿汇与二课等可扩展插件接入该项目,其中教务系统支持查询绩点、课表、考试成绩、选课分析与指导等功能,学习通支持作业查看与提交、签到等功能。该项目目前已尝试在校内推广使用,经反馈收集,该项目能很好地节省很多琐碎的操作,能够在 QQ 上直接与 AI 交互来完成一些功能。
总结: 这个项目让我系统理解了完整 Agent 的运行方式,包括上下文管理、系统提示词与 Skills 协调、提示词设计和记忆系统等,也让我在实际使用中持续观察模型容易犯错的场景,思考大模型与确定性程序之间的设计边界。
- **KCP 语言设计及编译器实现** 独立完成 [GitHub](#) 2026.03-至今
介绍: 该项目参考现代 C++ 与 Rust 编程语言的设计思想,自行设计了 KCP 语言的语法、类型系统和模块体系。基于现代 C++ 实现源码预处理、词法与语法分析、语义检查、IR 构建和 LLVM 代码生成,使 KCP 语言能够被编译、链接和执行。KCP 可与 C 语言混合编译链接。
同时,我实现了一套由 KCP 自身语言编写的标准库 std,包含字符串、动态数组、B 树、文件与 IO 等数据结构与算法。项目同时发布了 Linux 平台安装包与 CLion 插件,使这门语言能够真正地安装,并在 IDE 上编写与运行。KCP 及其标准库实现的一些大数据基准上与 C++、Rust 速度相当,甚至在一些基准上取得了更快的速度。相较 C++ 而言,舍弃了因兼容 C 的包袱导致遗存的各种语法问题。相较 Rust 而言,在类型安全上提供更多的自由度,编写体验接近于现代 C++。
总结: 这个项目让我系统理解了编译器各阶段的工作方式,掌握了语言解析、语义检查、IR 构建和代码生成之间的关系。
- **Q4_K 模型量化实现** 独立完成 [GitHub](#) 2026.06
介绍: 使用现代 C++ 从零实现 GGUF 纯 Q4_K 量化器,完成可量化权重的分块量化、尺度参数计算、低比特编码与 GGUF 写回,并正确处理元数据及非量化张量。在 Llama2-7B+LoRA 上实测,评估速度比官方 Q4_K_M 混合量化快约 50%,体积小约 7%,任务精度仅略低约 0.5%,甚至困惑度低约 10%,经去 LoRA 对照实验确认,官方把 LoRA 作用层用纯对称的 Q6_K 量化,无偏移补偿,而 LoRA 改变激活分布后误差被放大,导致困惑度提高。我的实现带 dmin 偏移补偿,对分布变化则更鲁棒。
总结: 这个项目让我意识到不存在一个通用且最好的量化算法,调优模型必须针对具体场景。
- **Encoder-Decoder Transformer 实现** 独立完成 [GitHub](#) 2026.06
介绍: 基于 PyTorch 从零实现可训练的 Encoder-Decoder Transformer,包含多头自注意力与交叉注意力、正弦位置编码、前馈网络、Pre-LN 残差结构及因果/填充掩码,并实现 KV Cache 增量解码。
- **JuXin: 现代 C++ 即时通信系统** 独立完成 [GitHub](#) 2025.11-2025.12
介绍: 为了学习协程、异步网络编程等技术,我基于 C++、Asio 与 MySQL 实现了包含服务端与客户端的即时通信系统。服务端围绕 Session 自主设计协程式异步架构,使用 strand、异步写队列与背压机制管理并发读写和慢客户端,并通过在线会话索引实现定向消息推送。以会话内递增序列号贯通消息排序、历史分页、已读状态与客户端缓存增量同步。
总结: 这个项目让我系统理解了网络通信与并发编程的基本原理,理解了客户端与服务端如何通过协议建立连接和交换数据,以及并发程序在任务调度、共享状态与执行顺序上和单线程程序的区别。

开源贡献

- **NVIDIA/stdexec: CMake 集成修复** [PR #1682](#) 2025.11-2025.12
我在编写现代 C++ 即时通信系统时引入了 stdexec 依赖,但发现 stdexec 与项目已有 Asio target 同时接入工程时出现 CMake target 冲突,为此我针对该问题向上游提交修复补丁,相关补丁已被合并。
- **UbiquitousLearning/mllm: Linux x86 构建与 FP16 量化修复** [PR #683 / #685](#) 2026.06
我计划为 mllm 贡献一套基于 C++ 的 Jinja 模板框架,因此尝试在我的 Linux x86 主机上运行该项目。我发现 MllmRT 共享库链接 xxHash 静态对象失败,在补充 PIC 构建配置并解决该问题后,又发现 x86 平台原有 FP16/FP32 转换将 FP16 对象直接传入只接受 16 位原始编码的 CPU 指令,触发隐式数值转换,使 FP16 数值先被转换为整数,再被当作位编码解析,最终导致数据错误。我增加了 FP16 raw bits 读写逻辑以保持原始位模式,修复这两处错误后,本机成功运行 mllm,为此向上游提交了两个修复补丁。
- **CMake/CMake: File API target graph 改进** [Issue #27890 / MR !12209](#) 2026.06
我在实现 Q4_K 模型量化时,在 C++ 代码中使用了 import std,此前能够正常构建的代码突然出现错误。对此我展开了问题排查,最终发现这是 CMake 上游更新引入的错误。为此我向上游报告问题,提交修复补丁并积极沟通解决方案,最终与上游共同完成相关重构与回归测试,相关补丁被 CMake 合并。
- **ggml-org/llama.cpp: LoRA 导出工具文档修复** [PR #24703](#) 2026.06
在完成 Q4_K 模型量化后,我需要部署 LoRA 模型测试量化效果。在使用 llama.cpp 的 export-lora 工具导出模型时,我发现文档中的 -lora-scaled 参数和多 LoRA 示例仍保留错误的旧信息,我修正文档并向上游提交补丁,相关补丁已被合并。
- **openai/codex: 插件缺陷报告** [Issue #18863 / Issue #36455](#) 2026.04-2026.08

在使用 codex 的插件生态时，发现本地插件安装丢失符号链接、Chrome 插件 API 契约与实现不一致等缺陷，逐一排查定位后向上游报告并给出修复方案

专业技能

- 熟练掌握现代 C++，具备扎实的 C 语言基础和三年 C++ 代码编写经验；熟悉 C++23 及部分 C++26 特性，理解面向对象、泛型与函数式编程范式，掌握 RAII、移动语义、Modules、Ranges、Concepts、模板元编程与 constexpr/constexpr 编译期计算、协程、异步 I/O、并发机制与内存模型。
- 熟悉数据结构与算法设计，掌握复杂度分析以及排序、查找、树、哈希、堆、图等常用算法与数据结构，理解分治、贪心与动态规划等算法思想；熟悉 C++ STL 容器、迭代器、Ranges 与算法库的设计与实现，理解 vector、deque、list、map/set、unordered_map、priority_queue 等组件的底层结构、复杂度保证和迭代器失效规则。
- 熟练使用 PyTorch，理解张量计算、自动微分及模型训练与推理机制，掌握 Transformer 与注意力模型，熟悉 KV Cache、LoRA、持续学习、知识蒸馏和模型量化等方法，具备模型实现、实验复现与训练问题定位能力。
- 熟悉计算机视觉与视觉表征学习，掌握 CNN/ResNet 特征提取、度量学习与图像检索，熟悉 FFT/傅里叶变换、幅度/相位分析、频带建模与频率调制，理解域泛化、持续学习与知识蒸馏，了解弱监督与跨模态视觉学习。
- 熟悉 x86/i386 汇编与 NASM，理解处理器启动、保护模式、中断与内存访问机制，掌握 GDT/IDT/TSS、PIC/PIT 等硬件机制，以及 MBR/Loader 引导、上下文切换和系统调用入口等底层实现。
- 以 Linux 作为主力操作系统，熟悉 Linux 桌面与命令行 workflow，掌握 Shell、dnf/RPM、Git、CMake/Ninja、GDB 等常用工具，熟悉软件安装配置、环境管理、项目构建调试和运行问题排查。
- 熟练使用 Qt 框架与 Qt Quick/QML，理解 Qt 元对象系统、信号与槽、事件循环等核心机制，掌握 C++/QML 协同、界面状态管理与网络交互。
- 熟悉 C++ 第三方库与依赖管理，熟悉 Boost.Asio、Boost.MySQL、nlohmann/json、stdexec 等库，了解 vcpkg、CPM.cmake、FetchContent、find_package 等依赖配置方式。
- 熟悉关系型数据库原理与应用，理解数据建模、约束、事务一致性与并发控制，熟悉 MySQL、PostgreSQL、数据库连接池管理。
- 熟悉 Python 语言及常用语法，掌握模块化与面向对象编程、类型标注，熟悉 NumPy、pandas、Matplotlib、SciPy 等数据处理、科学计算与可视化库，以及 FastAPI、asyncio、SQLite 等应用开发生态。
- 熟悉 AI Agent 运行时架构与工程实现，掌握 LLM 多模型适配、工具调用、结构化输出、上下文与记忆管理、任务编排、状态持久化与流式交互，熟悉数据 grounding、权限控制、审计、错误处理与可靠性验证，熟悉单智能体、多智能体与子代理协作机制。
- 熟练使用 Codex（累计 token 用量约 600 亿，花费 5000 元以上）、Claude Code 等 AI 编程代理，熟悉 Agentic 工作流与上下文工程，掌握代码库级理解、任务分解、上下文管理、迭代式实现与验证，熟悉 AGENTS.md、MCP、Skills、Hooks、插件等扩展机制，掌握 Git 驱动的代码实现、调试、重构、测试与审查流程。

社会实践

- **河北大学 ACM 校队** **队员/队长** 2024.06-2026.03
负责程序设计竞赛训练安排、题目讲解、答疑等，组织暑期线上、寒假线上和线下程序设计训练，为参加 ICPC、CCPC、天梯赛、蓝桥杯等程序设计竞赛做准备。
- **第二届保定市高校四校程序设计联赛** **赛事主要负责人** 2025.10-2025.12
负责联赛组织推进、校际沟通协调和赛事流程安排，为河北大学、华北电力大学（保定）、河北农业大学、保定学院四校学生搭建程序设计竞赛交流平台。
- **首届河北大学程序设计竞赛暨新生赛** **赛事组织者** 2025.10-2025.11
筹办首届河北大学程序设计校赛，吸引校内 18 个院系、覆盖文理工医等多学科的 103 支有效参赛队、300 余名学生参赛。参与命题、验题、场地协调和现场流程设计，组织气球奖励、滚榜展示等竞赛环节，增强赛事仪式感与校内程序设计竞赛影响力。
- **计算机文化节——AI 沙龙与辩论赛** **活动主要负责人** 2025.10-2025.11
负责河北大学第九届计算机文化节 AI 沙龙与辩论赛活动组织，围绕 AI 工具在资料检索、数据处理、思路拓展等学习场景中的应用开展实践分享，并组织“AI 工具进入大学课堂利弊”主题辩论，促进跨专业学生对 AI 应用方法、学术规范和技术边界的交流讨论。

自我评价

- **对计算机保持长期兴趣**：对计算机领域的各种知识保持好奇，喜欢动手折腾和尝试，习惯从基本原理出发理解问题，并在实践中验证想法、持续改进。**做事认真负责**，是我做的我就会努力争取最好的结果，即便是失败了我也会继续分析原因并改进。
- **具备较强的自主学习能力**：面对陌生问题时，我会主动检索论文和相关资料，根据问题快速建立自己的认知与体系，围绕解决问题去自主探索方法。**吃苦耐劳**，上课认真，课余时间则精心投入自己的事情，经常学到很晚，写论文期间时常选择通宵根据老师意见修改论文。寒暑假期间我都会要求自己额外学习一个计算机技能，我尝试过游戏制作、视频剪辑、写操作系统等。
- **未来规划**：保持自己的兴趣、提升自己的科研与工程能力，解决问题的能力。认真投入工作、提升自己。希望我能在我的领域形成我自己的理解，建立我自己的体系，从学习模仿到自主创新，做出有价值的成果。