

# CXXCrafter: An LLM-Based Agent for Automated C/C++ Open Source Software Building

CXXCrafter: 一种基于大语言模型的 C/C++ 开源软件自动化构建代理

汇报人

于康朔

指导老师

文成



...optimized, portable, and performance-oriented.  
Building C/C++ software [53] involves many steps, including dependency management, managing platform-specific challenges. These processes are critical for producing robust, portable, and high-quality software. Such tasks require

简介中“构建 C/C++ 软件 [53] 包括编译、解决依赖关系、链接库、配置环境以及应对特定平台的挑战。这些过程对于执行自动化程序分析至关重要，尤其是动态分析。”是对 C/C++ 项目构建过程的概括性描述

## 应对特定平台的挑战 (Platform-Specific Challenges)

处理不同操作系统、硬件架构、编译器差异等带来的兼容性问题。

## 链接库 (Linking Libraries) 三

将编译后的目标文件与所需的静态库 (.a/.lib) 或动态库 (.so/.dll) 合并，生成可执行文件或共享库

## 编译 (Compilation)

将源代码 (.c/.cpp) 转换为目标代码 (.o/.obj)

## 配置环境 (Environment Configuration)

设置编译环境变量、路径、工具链版本等。

## 解决依赖关系 (Dependency Resolution)

识别并获取项目所依赖的外部库或模块，比如用于加密的openssl库，图像处理的opencv库，构建系统需要找到并准备好这些库

# 2

## 实证研究

---

- 研究设计
- 关键发现
- 总结三大核心挑战
- 研究方案可行性说明



## 2.1 实证研究

第 10 页

### 数据集

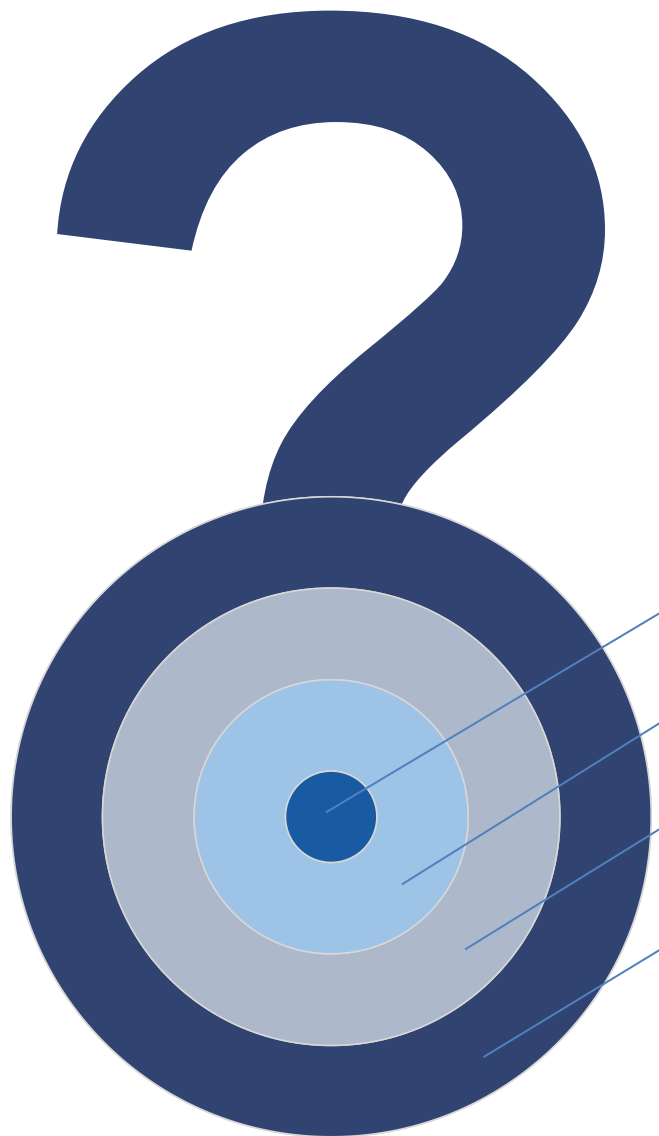
Top100 热门 C/C++ 项目（覆盖 10 类：内核工具、游戏开发、AI 框架等，表1）

Table 1: The Top 100 Projects and Their Categories

Rank	Project
1	Linux Kernel
2	Redis
3	TensorFlow
4	PyTorch
5	Apache Spark
6	Apache Flink
7	Apache Hadoop
8	Apache Kafka
9	Apache Storm
10	Apache Pig
11	Apache Tez
12	Apache Mahout
13	Apache Avro
14	Apache Thrift
15	Apache Parquet
16	Apache Iceberg
17	Apache Delta Lake
18	Apache Hudi
19	Apache Kudu
20	Apache Druid
21	Apache Solr
22	Apache Elasticsearch
23	Apache OpenSearch
24	Apache Lucene
25	Apache Nutch
26	Apache Mahout
27	Apache Hama
28	Apache Storm
29	Apache Flink
30	Apache Spark
31	Apache Hive
32	Apache Pig
33	Apache Tez
34	Apache Mahout
35	Apache Avro
36	Apache Thrift
37	Apache Parquet
38	Apache Iceberg
39	Apache Delta Lake
40	Apache Hudi
41	Apache Kudu
42	Apache Druid
43	Apache Solr
44	Apache Elasticsearch
45	Apache OpenSearch
46	Apache Lucene
47	Apache Nutch
48	Apache Mahout
49	Apache Hama
50	Apache Storm
51	Apache Flink
52	Apache Spark
53	Apache Hive
54	Apache Pig
55	Apache Tez
56	Apache Mahout
57	Apache Avro
58	Apache Thrift
59	Apache Parquet
60	Apache Iceberg
61	Apache Delta Lake
62	Apache Hudi
63	Apache Kudu
64	Apache Druid
65	Apache Solr
66	Apache Elasticsearch
67	Apache OpenSearch
68	Apache Lucene
69	Apache Nutch
70	Apache Mahout
71	Apache Hama
72	Apache Storm
73	Apache Flink
74	Apache Spark
75	Apache Hive
76	Apache Pig
77	Apache Tez
78	Apache Mahout
79	Apache Avro
80	Apache Thrift
81	Apache Parquet
82	Apache Iceberg
83	Apache Delta Lake
84	Apache Hudi
85	Apache Kudu
86	Apache Druid
87	Apache Solr
88	Apache Elasticsearch
89	Apache OpenSearch
90	Apache Lucene
91	Apache Nutch
92	Apache Mahout
93	Apache Hama
94	Apache Storm
95	Apache Flink
96	Apache Spark
97	Apache Hive
98	Apache Pig
99	Apache Tez
100	Apache Mahout



## 2.2 关键发现



RQ1: 默认构建成功率仅 21% (100 个项目仅 21 个成功, 表 2), 79% 项目需额外配置

RQ2: 构建失败核心原因 (表 4):

1. 库相关问题 (284 次, 占 73.9%): 库未安装、路径错误、版本冲突
2. 配置问题: 系统不兼容、构建命令错误、项目自定义配置缺失

3. 构建工具链问题 (64 次): 工具缺失、版本不兼容 (如 CMake 版本需求不匹配)

4. 其他问题: 内存问题, 源代码问题



## 2.2关键发现

RQ2

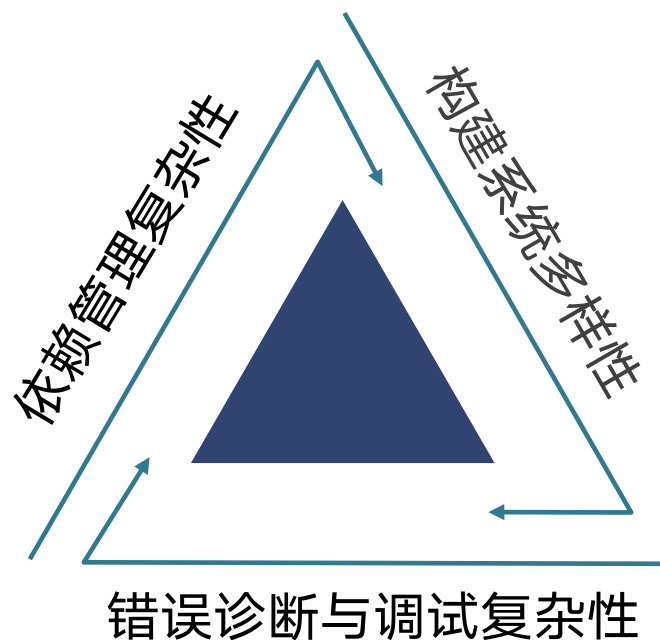
Table 4. Most Significant Social Issues for the Building of the City (RQ2) Perceived by Interviewees

Category	Social Issues	Frequency	Comments
Economic issues	Unemployment	14	Unemployment is a major problem in the city, especially for young people.
	Income inequality	10	Income inequality is a major problem in the city, especially for young people.
	Overcrowding	10	Overcrowding is a major problem in the city, especially for young people.
Social issues	Healthcare	14	Healthcare is a major problem in the city, especially for young people.
	Education	10	Education is a major problem in the city, especially for young people.
	Environment	10	Environment is a major problem in the city, especially for young people.
Cultural issues	Language	14	Language is a major problem in the city, especially for young people.
	Religion	10	Religion is a major problem in the city, especially for young people.
	Traditions	10	Traditions is a major problem in the city, especially for young people.
Political issues	Corruption	14	Corruption is a major problem in the city, especially for young people.
	Human rights	10	Human rights is a major problem in the city, especially for young people.
	Democracy	10	Democracy is a major problem in the city, especially for young people.
Environmental issues	Climate change	14	Climate change is a major problem in the city, especially for young people.
	Pollution	10	Pollution is a major problem in the city, especially for young people.
	Resource depletion	10	Resource depletion is a major problem in the city, especially for young people.
Other issues	Gender inequality	14	Gender inequality is a major problem in the city, especially for young people.
	Disability	10	Disability is a major problem in the city, especially for young people.
	Ageism	10	Ageism is a major problem in the city, especially for young people.



## 2.3总结三大核心挑战

第 15 页



“

- 依赖管理复杂性：外部库多，包管理工具（Conan、vcpkg）支持库集与使用模式差异大。
- 构建系统与编译选项多样性：20 + 种构建系统（Makefile、CMake 等），语法与配置要求各异
- 错误诊断与调试复杂性：多阶段（预处理、编译、链接）错误，信息差异大，难统一处理

”

# 3

## 方法论：CXXCrafter 系统设计

---

- 设计理念
- 三大核心模块
- 提示设计细节







# 三大核心模块

第 18 页



解析器  
模块

**功能：提取项目关键信息，为构建提供基础数据**

环境信息提取器：lscpu/uname -a 获取 CPU、OS 版本，解决系统兼容性问题  
依赖信息提取器：结合 CCScanner 静态识别依赖名与版本，初步规避依赖错误  
文档信息提取器：正则筛选 + LLM 精筛构建相关文档（README、COMPILE），用 RAG 检索关键指导



生成器  
模块

**核心输出：基于解析信息生成 Dockerfile（含 shell 构建脚本）**

动态优化：接收执行器错误反馈，在同一会话中保留修改历史，超上下文限制时清除旧问题  
提示设计：嵌入式模板（含环境、依赖、文档占位符），确保 Dockerfile 结构标准化（图 4）



执行器  
模块

**环境管理：保证环境干净可重现**

错误处理：捕获日志并反馈给生成器，形成迭代循环（最大 10 次交互）  
LLM 判别器：解决“非错误失败”（如自定义错误抑制、指令缺失），基于静态（含 make/cmake 指令）+ 动态（日志编译进度）标准判断构建成功



提示设计  
细节

**5 种场景专属提示**

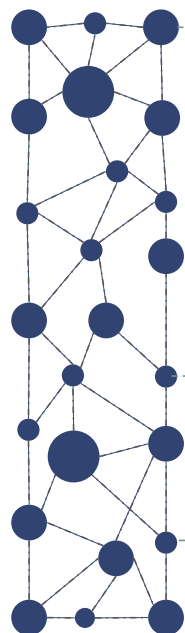
构建系统识别、文档 RAG 解析、初始 Dockerfile 创建、Dockerfile 修改、构建成功判别  
挑战应对：嵌套模板分解复杂问题、RAG 补充项目细节、上下文管理解决标记限制

# 4

## 评估：CXXCrafter 性能验证

---

- 评估设计
- 整体有效性
- 组件消融分析
- 案例研究



## 数据集

Top100 项目 + Awesome-CPP (652 个项目, 共 752 个), 剔除了两个数据集的相同项目

## 基线对比

默认构建命令、裸 LLM (DeepSeek-v2/v3、GPT-4o/mini)、人工构建

## 成功标准

静态 (Dockerfile 含必要指令) + 动态 (日志无错误), 人工验证确保准确性



## 整体有效性（回答RQ3）

第 21 页

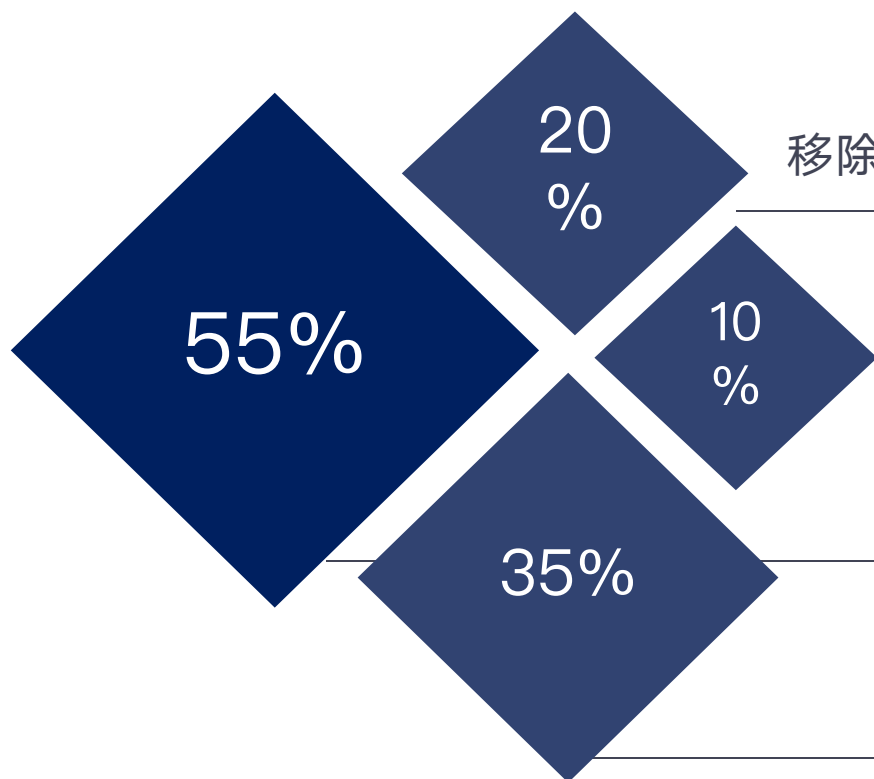
**RQ3（有效性）：CXXCrafter 能自动构建多少个 C/C++ 项目？**

CXXCrafter（默认 GPT-4o）成功率 78.1%：Top100 成功 75 个，Awesome-CPP 成功 512 个  
显著优于基线：默认构建命令（39.01%）、裸 LLM（最高 38.43%），人工构建成功 86 个  
（但 CXXCrafter 节省人力）

Table 1: Experimental Results Between CXXCrafter and Baselines.

Metric/Category	Top100 Build	Awsome-CPP Build	Total Build	Success Rate (%)
Programmer	86	512	418	21.00
Default Build Command	39	212	178	39.01
Base LLM (DeepSeek-V2)	41	212	180	41.42
Open LLM (DeepSeek-V2)	42	212	180	42.00
Base LLM (GPT-4o-mini)	47	212	180	47.78
Base LLM (GPT-4o)	52	212	180	51.43
CXXCrafter (DeepSeek-V2)	62	211	177	62.71
CXXCrafter (Default)	75	512	547	78.48

注：表中“Total Build”指成功构建的项目总数，而非尝试构建的项目总数。CXXCrafter 的成功率显著高于基线，表明其在自动构建 C/C++ 项目方面具有更强的能力。



## 解析器的必要性

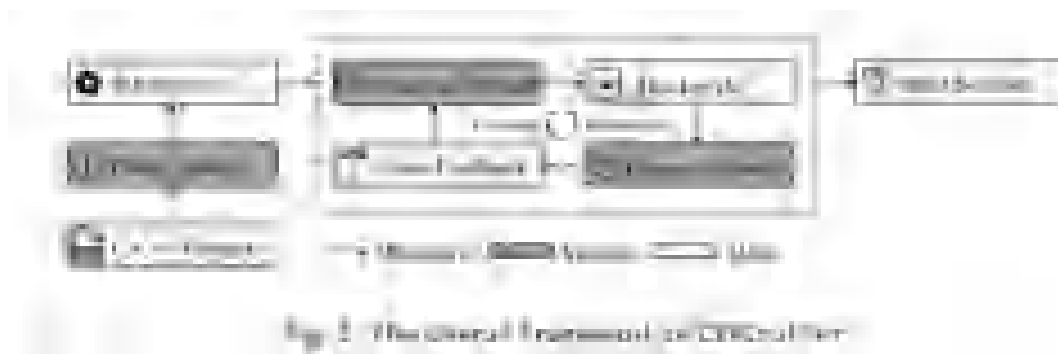
移除后成功数从 75 降至 48，因缺失文档解析与依赖识别

## 动态交互的重要性

禁用后成功数降至 22；10 步交互最优  
(5 步 69 个、20 步 74 个，超阈值收益递减)

## LLM 影响

GPT-4o 最优 (75 个) > DeepSeek-v3 (67 个)  
> DeepSeek-v2 (57 个) > GPT-4o mini (50 个)





RQ5: CXXCrafter 是如何解决手动方法无法解决的构建问题的，它在处理复杂的 C/C++ 项目时具有哪些具体优势？

人工案例  
失败分析

3 个人工失败案例分析：

CQuery：人工因 Clang 版本兼容失败，CXXCrafter 推荐 Ubuntu 20.04 解决

Paddle：人工忽略 protobuf 与 oneDNN 版本匹配，CXXCrafter 从 requirements.txt 识别

DOOM：人工误判 32 位库问题，CXXCrafter 定位 TSL 变量与共享库版本不匹配

CXXCrafter  
or  
优势

CXXCrafter 优势：RAG 高效文档处理、LLM 隐性历史知识补充。