·2026中国汽车工程学会年会优秀论文(选登)·

基于大语言模型的智能座舱应用缺陷自动分析方法

刘杰　郑红丽　陈彩可　安星霖

(中国第一汽车股份有限公司研发总院,长春 130013)

【欢迎引用】刘杰, 郑红丽, 陈彩可, 等. 基于大语言模型的智能座舱应用缺陷自动分析方法[J]. 汽车文摘, 2026(10): 1-8.

【Cite this paper】LIU J, ZHENG H L, CHEN C K, et al. A Large Language Model Based Approach for Automatic Defect Analysis of Intelligent Cockpit Applications[J]. Automotive Digest (Chinese), 2026(10): 1-8.

【摘要】针对智能座舱应用缺陷分析效率低下的问题,提出一种基于大语言模型(LLM)的四阶段自动分析框架CockpitInsight。该框架包括缺陷多模态理解、日志自适应过滤、上下文增强和链式思维根因推理4个阶段。其中,自适应时间窗口过滤算法通过核心区、扩展区、背景区三区域分层策略,将17万行原始日志压缩至约700行(压缩比99.6%),解决了大体量日志与LLM上下文窗口不匹配的问题;LLM摘要增强的语义检索方法通过对历史缺陷生成结构化摘要消除表述差异,在1 054条缺陷索引上实现了高质量的跨项目语义匹配。试验数据表明,端到端分析耗时70 s(含视频分析),生成报告包含完整推理链路和日志证据引用,验证了LLM驱动的多模态缺陷分析在智能座舱领域的工程可行性。

关键词:智能座舱;缺陷分析;大语言模型;日志分析;根因定位;语义检索;视觉语言模型

中图分类号:U463.6　　**文献标志码:**A　　**DOI:** 10.19822/j.cnki.1671-6329.20260169

A Large Language Model Based Approach for Automatic Defect Analysis of Intelligent Cockpit Applications

Liu Jie, Zheng Hongli, Chen Caike, An Xinglin

(Global R&D Center, China FAW Corporation Limited, Changchun 130013)

【Abstract】Aiming at the low efficiency of defect analysis for intelligent cockpit applications, this paper proposes CockpitInsight, a four-stage automatic analysis framework based on Large Language Models (LLM). The framework comprises multimodal defect understanding, adaptive log filtering, context augmentation, and Chain-of-Thought root cause reasoning. The adaptive time-window filtering algorithm employs a core, extended and background three-zone layered strategy to compress 170 000 raw log lines to approximately 700 lines (99.6% compression ratio), resolving the contradiction between massive logs and LLM context window limitations. The LLM-summary-enhanced semantic retrieval method generates structured summaries for historical defects to bridge expression gaps, achieving high-quality cross-project semantic matching on an index of 1 054 defects. Validation data shows an end-to-end analysis time of 70 s (including video analysis), with generated reports containing complete reasoning chains and log evidence references, demonstrating the engineering feasibility of LLM-driven multimodal defect analysis in the intelligent cockpit domain.

Key words: Intelligent cockpit, Defect analysis, Large language model, Log analysis, Root cause localization, Semantic retrieval, Vision-language model

0　引言

随着智能网联汽车的快速发展,智能座舱已成为汽车差异化竞争的核心领域[1]。现代座舱系统基于车载Android平台构建,每款车型搭载15~30个应用,座舱应用的缺陷数量随之增加。以某整车制造企业(Original Equipment Manufacturer, OEM)的实际生产为例,每个座舱模块每月平均上报缺陷高达50~200个。当前缺陷分析仍以人工模式为主,全流程平均耗时2~4 h。其中,日志筛查耗时最长,单个缺陷的日志文件约17万行,有效信息仅数十行,信噪比极低。此外,约30%~40%的缺陷与历史已解决问题根因相同,但由于缺乏语义检索手段,工程师需对同类问题重复分析。

大语言模型(Large Language Model,LLM)在自然语言理解和推理任务上展现出强大能力[2]。检索增强生成(Retrieval-Augmented Generation,RAG)技术[3]和思维链(Chain of Thought,CoT)推理[4]拓展了LLM在知识密集型任务中的应用。然而,将LLM应用于座舱缺陷分析时,日志体量远超上下文窗口,缺陷描述存在大量模版噪声,且需融合多模态信息。

针对上述问题,本文提出了基于LLM的智能座舱缺陷自动分析框架CockpitInsight。该框架采用四阶段流水线架构:(1)缺陷多模态理解:模版噪声清洗、LLM结构化信息提取,以及视觉语言模型(Vision-Language Model,VLM)视频分析与自动语音识别(Automatic Speech Recognition,ASR),从复现视频中提取屏幕时间戳,为后续日志过滤提供精确时间锚点;(2)日志自适应过滤:三区域分层过滤策略,将17万行日志压缩至约700行(压缩比99.6%);(3)上下文增强:LLM摘要增强的FAISS(Facebook AI Similarity Search)语义检索;(4)根因推理:CoT提示策略综合多源证据生成结构化报告。验证表明端到端分析耗时约70 s,报告包含完整推理链路和日志证据引用。本文的主要贡献包括:(1)提出四阶段LLM驱动缺陷分析框架,创新性地将VLM视频分析与ASR语音识别融入缺陷理解阶段,实现视频屏幕时间戳到日志时间窗口的精确对齐;(2)设计自适应时间窗口日志过滤算法;(3)构建LLM摘要增强的语义缺陷检索机制;(4)在真实OEM数据上进行端到端验证。

1 相关研究概述

1.1 日志智能分析

日志异常检测是AIOps领域的核心研究方向。Du等[5]提出DeepLog模型,利用长短期记忆(Long Short-Term Memory,LSTM)网络检测日志异常。Meng等[6]提出LogAnomaly模型,实现无监督异常检测。He等[7]通过Drain算法进行在线日志模版提取,并利用LLM进行日志解析和故障诊断[8]。

然而,上述工作主要面向云原生场景,侧重于异常检测而非根因分析。智能座舱环境的日志为Android logcat格式,单次缺陷日志量较大,且涉及多应用交互。本文提出的自适应时间窗口过滤策略通过三区域分层过滤实现99.6%的压缩比。

1.2 LLM在软件缺陷分析中的应用

在传统缺陷分析领域,Lamkanfi等[9]开展了缺陷严重度预测研究,基于文本特征实现Bug报告的自动分类。重复缺陷检测(Duplicate Bug Report Detection,DBRD)[10]通过文本相似度计算帮助识别重复提报的缺陷。然而,上述方法主要基于词频统计或浅层语义特征,难以捕捉不同表述方式的深层语义等价关系。

LLM为软件工程带来了新范式。Xia等[11]利用LLM实现自动程序修复。Lewis等[3]提出RAG技术框架,Wei等[4]提出CoT策略提升多步推理能力。然而,RAG系统在检索-生成交互中引入了新形式的幻觉,Zhao和Elchafei的研究从证据归因层面进行缓解[12-13]。在AIOps领域,文献[14]基于LLM进行了根因分析,但尚无面向车载座舱缺陷的系统性研究。

本文将LLM和CoT技术综合应用于智能座舱缺陷分析。与现有工作相比,本文面向多模态数据(工单、日志、视频),覆盖全链路分析,并创新性地引入LLM摘要增强语义检索。

1.3 汽车软件质量保障

汽车软件过程改进及能力评定(Automotive Software Process Improvement and Capability Determination,ASPICE)和ISO 26262为汽车软件质量提供了流程规范框架[15-16]。现有方法以流程规范和人工审查为主,缺陷分析环节缺乏AI自动化手段。本文首次将LLM驱动的多模态缺陷分析应用于智能座舱领域。

2 问题定义与需求分析

2.1 缺陷分析工作流

在智能座舱缺陷分析实践中,开发人员需依次完成:阅读描述(约5 min)、下载日志(10~30 min)、搜索历史问题(10~30 min)、日志定位异常(30~60 min)、根因推理(30~60 min)以及编写报告(15~30 min)。全流程平均耗时2~4 h,如图1所示。其中,日志分析和根因推理占总耗时的60%以上。

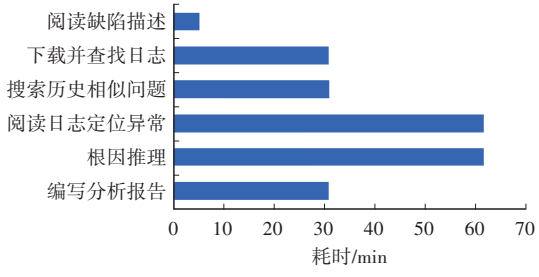


图1 人工缺陷分析工作流耗时分布

2.2 用户需求调研

为深入理解一线开发人员的实际痛点,本研究对

某OEM智能座舱开发部门的6位开发负责人进行了结构化访谈，如图2所示。访谈内容覆盖电话/蓝牙、空调和座椅、手车互联、桌面和Launcher、空中下载技术（Over The Air，OTA）和设置、综合管理等模块，团队规模为3~10人，日均处理缺陷数量为3~20个。

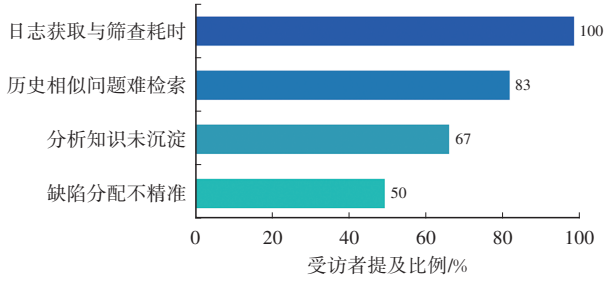


图2　开发人员痛点调研结果

访谈揭示了3个核心痛点：（1）日志筛查耗时：6位受访者（100%）均将其列为最大痛点，30 MB以上的日志文件人工筛查耗时0.5~6 h；（2）相似问题难检索：5位（83%）反映不同工程师对同一问题表述差异大，关键词搜索难以有效匹配；（3）知识沉淀不足：4位（67%）指出经验存储依赖个人记忆，新人学习成本较高。

2.3　问题形式化定义

基于上述分析，本文将智能座舱缺陷自动分析问题分解为以下3个子问题。

（1）定义1（缺陷日志过滤问题）：给定缺陷工单 $\boldsymbol{B}=\{\text{title, desc, metadata}\}$ 及其关联的原始日志 $\boldsymbol{L}(|\boldsymbol{L}|\sim10^5$行），目标是生成精炼日志切片 $\boldsymbol{S}_B\subset\boldsymbol{L}(|\boldsymbol{S}_B|\sim10^2$行），使得与问题相关的关键事件链 $\boldsymbol{E}=\{e_1,\cdots,e_k\}$ 在切片中完整保留，即 $\boldsymbol{E}\subseteq\boldsymbol{S}_B$。

（2）定义2（相似缺陷检索问题）：给定当前缺陷 B_q 及历史缺陷库 $\boldsymbol{D}=\{\boldsymbol{B}_1,\cdots,\boldsymbol{B}_m\}$（每条缺陷含LLM生成的结构化摘要 z_i），目标是返回按语义相似度排序的Top-K列表 $\boldsymbol{R}=\{(\boldsymbol{B}_i,s_i)\}$，其中 $s_i=\cos\left(\text{Enc}\left(z_q\right),\text{Enc}\left(z_i\right)\right)$ 为基于文本嵌入模型Enc(，)的余弦相似度。

（3）定义3（根因定位问题）：给定缺陷描述 $\boldsymbol{B}$、精炼日志 $\boldsymbol{S}_B$、相似历史 $\boldsymbol{R}$ 及视频描述 $\boldsymbol{V}$，目标是生成结构化根因分析报告 $\boldsymbol{A}=\{\text{cause, module, severity, fix, evidence, confidence}\}$。

2.4　关键挑战

实现上述目标面临以下挑战：（1）日志信噪比极低，17万行中仅约50行为关键事件；（2）存在语义鸿沟，相同问题表述差异巨大，关键词匹配失效；（3）缺陷描述超70%为模版噪声；（4）30 MB日志远超LLM上下文窗口；（5）结构化工单、日志文本和视频需联合分析。

3　方法设计

3.1　整体架构

本文提出的CockpitInsight框架采用四阶段流水线架构，如图3所示。4个阶段分别为：缺陷多模态理解（Stage 1）、日志自适应过滤（Stage 2）、上下文增强（Stage 3）和根因推理（Stage 4）。

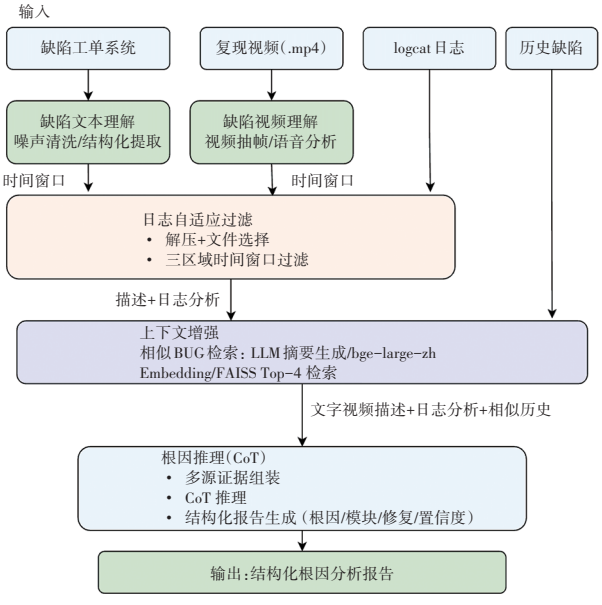


图3　CockpitInsight四阶段分析架构

该架构的核心设计原则是确定性计算与LLM推理的解耦。Stage 1为多模态LLM推理阶段，融合文本理解与VLM视频分析；Stage 2为确定性计算阶段，利用Stage 1提取的时间点（含视频屏幕时间戳）将日志压缩到LLM可处理范围；Stage 3为向量检索阶段；Stage 4为CoT根因推理阶段。

输入包含3类数据源：（1）缺陷工单信息（35个字段）；（2）Android logcat日志（约31 MB/17万行）；（3）复现视频。输出为Markdown和JSON格式的根因分析报告。

3.2　缺陷多模态理解

缺陷多模态理解阶段的目标是从原始缺陷工单和复现视频中提取结构化信息，为后续阶段提供语义基础和时间锚点。该阶段包含2个子阶段：缺陷文本理解（Stage la）和视频场景理解（Stage lb）。

缺陷文本理解阶段包括3个步骤：（1）字段映射标准化：由于缺陷管理系统存在中英文字段名混用

的问题(如问题标题和Title指向同一字段),本文设计了18组字段映射规则,将原始字段统一映射为标准名称。(2)模版噪声清洗:实际缺陷描述中包含大量固定模版内容,严重干扰LLM的信息提取。本文设计了8条正则表达式清洗规则,分别用于移除开发回复模版、测试回复模版、monitor日志模版、统一资源定位符(Uniform Resource Locator,URL)链接等噪声内容。经清洗后,描述字段的有效信息占比由不足30%提升至90%以上。(3)LLM结构化信息提取:将清洗后的标题和描述输入LLM,引导模型输出问题现象、复现步骤、问题时间点、环境信息、缺陷分类、关键词列表和初步假设等信息。其中,问题时间点的提取将与视频屏幕时间戳共同驱动日志时间窗口定位。当工单文本中缺少精确时间信息时,缺陷视频理解阶段提取的屏幕时间戳可作为关键补充。(3)视频场景理解:针对附带复现视频的缺陷,本文设计了多模态视频分析流程。首先从视频中均匀抽取关键帧,对每帧独立调用VLM(Qwen3-VL云端部署)通过座舱UI分析提示词描述界面布局和异常情况;同时提取音频轨道并通过ASR进行语音识别,获取测试人员的口述描述。VLM逐帧描述与ASR转录文本由文本LLM综合推理,生成包含异常时间点、问题现象和屏幕时间戳的视频分析报告。其中,屏幕时间戳从视频画面中车载系统的时钟显示提取,作为精确时间锚点反馈至Stage 2,使日志过滤能够在缺少工单时间信息时仍准确定位到问题发生的时间窗口。

3.3 日志自适应过滤

日志自适应过滤是本框架的核心技术创新之一,旨在解决大体量日志与LLM上下文窗口之间的矛盾。如图4所示,本文提出自适应时间窗口过滤算法,采用核心区(Core)、扩展区(Extended)和背景区(Background)三区域分层策略。

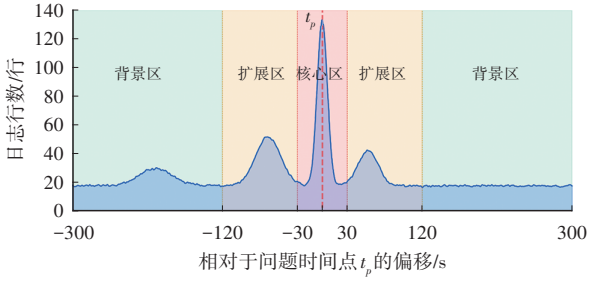


图4 自适应时间窗口日志过滤策略

算法描述:给定原始日志文件F(|F|~1.7 × 10^5行)、Stage 1提取的问题时间点列表$\boldsymbol{T}=\{t_1,\cdots,t_n\}$、Tag白名单$\boldsymbol{W}_{\text{tag}}$和关键词列表$\boldsymbol{W}_{\text{kw}}$,算法输出精炼日志切片$\boldsymbol{S}_B$。对每个问题时间点$t_p\in\boldsymbol{T}$,3个区域的处理策略如下。

(1)核心区$\left(\left|t_i-t_p\right|\leqslant 30\text{ s}\right)$:保留全部日志行,不做任何过滤。该区域覆盖问题触发的关键瞬间,确保事件链的完整性。

(2)扩展区$\left(30\text{ s}<\left|t_i-t_p\right|\leqslant 2\text{ min}\right)$:仅保留日志Tag属于白名单$\boldsymbol{W}_{\text{tag}}$的行。白名单包含与当前缺陷模块相关的关键Tag(如ActivityTaskManager、Unity渲染相关Tag等),过滤掉无关的系统噪声日志。

(3)背景区$\left(2\text{ min}<\left|t_i-t_p\right|\leqslant 5\text{ min}\right)$:仅保留包含关键词列表$\boldsymbol{W}_{\text{kw}}$中关键词(如error、exception、crash、fatal等)的行,用于捕获可能的背景异常。

3个区域的结果按时间排序合并后,执行行级去重操作(移除连续完全重复的日志行),最终得到精炼日志切片。在BUG-10924747的验证中,169 059行原始日志经3个区域过滤后得到6 259行,再经去重得到713行,总压缩比达99.6%。

此外,Stage 2还自动生成日志分析辅助信息,包括关键事件时间线(按时间排列的主要事件摘要)、Error/Warning级别日志汇总以及异常模式识别结果,为Stage 4的根因推理提供结构化的日志证据。

3.4 上下文增强

上下文增强通过历史缺陷语义检索为根因推理提供上下文信息。

相似缺陷语义检索:本文提出LLM摘要增强的语义检索方法,如图5所示,包含离线索引构建和在线检索2个阶段。

在离线阶段,对历史缺陷库$\boldsymbol{D}$中每条缺陷利用LLM生成结构化摘要(含现象、原因、关键词),将标题与摘要拼接后通过BGE-Large-ZH编码为1 024维向量,存入FAISS向量索引。共对1 054条导航缺陷完成索引构建。

LLM摘要增强的核心价值在于消除表述差异。不同工程师对同一问题的描述差异显著,直接嵌入编码会导致语义匹配失效。LLM摘要将描述统一为"现象/原因/关键词"格式,显著提升了语义检索质量。

在线检索阶段,对当前缺陷同样生成LLM摘要并编码为向量,在FAISS索引上执行Top-4最近邻检索,返回相似度最高的4条历史缺陷及其关联信息(标题、模块、状态、LLM摘要、缺陷管理系统链接)。

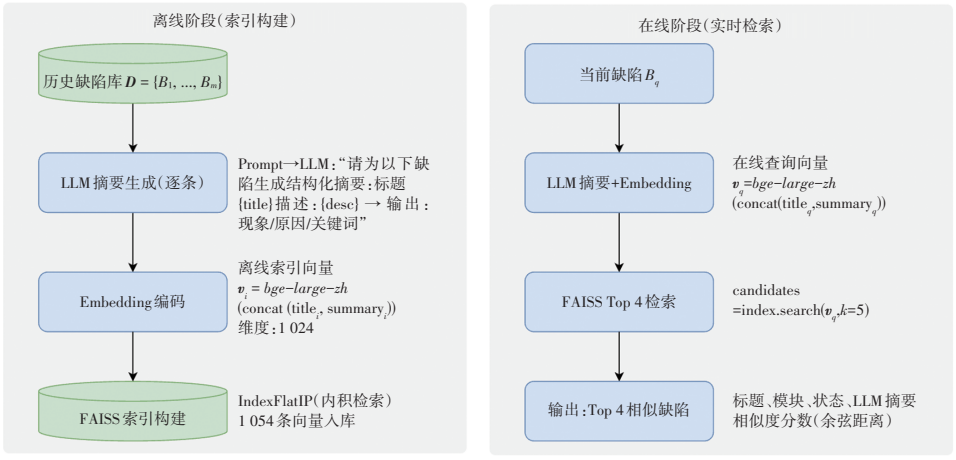


图5　LLM摘要增强的语义缺陷检索流程

根因推理优先以当前缺陷的日志证据和视频描述为首要依据,历史Top-K候选仅提供参考支撑。系统为每条待分析缺陷建立独立上下文,所有历史候选均通过唯一缺陷编号标注来源,不与本身缺陷的信息混用。

3.5　根因推理

根因推理将前3个阶段的输出作为多源证据进行组装,利用LLM的CoT推理能力生成结构化根因分析报告。

证据组装:输入LLM的上下文由4部分组成:(1)缺陷文本理解阶段输出的结果,如现象、分类、假设、关键词;(2)缺陷视频理解阶段输出的视频描述与ASR转录(若有);(3)Stage 2输出的日志证据(关键事件时间线、异常模式、Error/Warning汇总);(4)Stage 3输出的相似历史Top-4(标题、模块、LLM摘要)。

CoT推理:提示词引导LLM按7个步骤逐步推理:Step 1从日志中识别异常事件链;Step 2结合缺陷描述确认问题现象;Step 3参考相似历史缺陷的解决方案;Step 4若有视频,则对比视频中观察到的UI状态;Step 5推理根本原因;Step 6判定责任模块和组件;Step 7提出修复建议。

结构化输出:LLM以JSON格式输出根因分析报告,包括问题摘要、根因分析(包括推理链路和日志证据)、责任方判定、严重程度评估、修复建议、置信度评估及信息缺口说明,并自动转换为Markdown可读报告。

3.6　工程实现

CockpitInsight采用Python 3.13实现。LLM推理采用双后端架构:使用云端30 B参数模型,不可用时降级到本地7 B参数模型(Ollama框架)。VLM采用Qwen3-VL视觉语言模型云端部署,ASR采用Paraformer-realtime-v2部署。所有LLM调用配置指数退避重试机制,确保稳定性。

4　验证与评估

4.1　试验设置

数据集:试验数据来源于某OEM智能座舱缺陷管理系统。历史索引库含1 054条导航缺陷(FAISS向量索引)。验证样本为BUG-10924747(分屏下切换应用时车模消失),包含缺陷系统描述信息、3个日志文件(每个约31 MB/17万行)和2个视频。该缺陷已关闭,评论中包含完整根因讨论,可作为人工分析的Ground Truth。

模型配置:如表1所示,文本LLM采用30 B参数的开源模型(AWQ 4bit量化,云端GPU部署),文本嵌入采用BGE-Large-ZH模型(1 024维,本地CPU推理),VLM采用Qwen3-VL在线调用(云端部署),ASR采用Paraformer-realtime-v2(云端部署),向量索引采用FAISS IndexFlatIP。

表1　模型与硬件配置

模块/组件	模型与技术选型	配置与部署说明
文本LLM	30 B参数开源模型	AWQ 4bit量化,云端GPU部署
文本嵌入模型	BGE-Large-ZH模型(1 024维)	本地CPU推理
VLM	Qwen3-VL视觉语言模型	云端部署
向量索引	FAISS indexFlatIP	

4.2　案例分析

以BUG-10924747为例展示分析过程,并与人工分析对比。

Stage 1结果:在缺陷文本理解阶段,LLM正确提取了缺陷的核心信息。现象为"在分屏状态下切换应用时车模消失",问题时间点为15:36和15:40(与日志

记录吻合），缺陷分类为UI异常，初步假设为“分屏模式下界面渲染逻辑异常导致车模被错误隐藏”。所有关键字段提取准确。在缺陷视频理解阶段，对2个复现视频进行了逐帧VLM分析与ASR语音识别，从视频画面中提取到屏幕时间戳并识别出问题异常时刻，为Stage 2日志过滤提供了精确的时间锚点。

Stage 2结果：如图6所示，169 059行原始日志经自适应时间窗口过滤后得到6 259行（压缩96.3%），再经行级去重后得到713行（总压缩99.6%）。AI提取的关键事件时间线准确覆盖了问题发生的完整过程：15:35:18进入分屏模式，15:37:09 Unity渲染服务状态变更，15:37:10 Launcher可见性通知，15:37:12–15:37:13导航主活动生命周期异常。

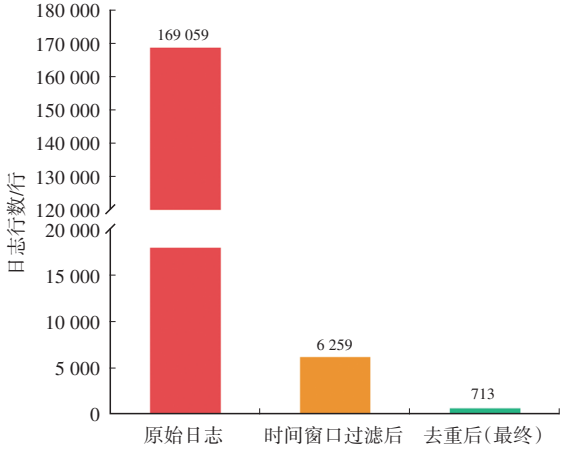


图6　日志过滤各阶段效果

如表2所示，FAISS语义检索返回的Top-4结果全部为分屏模式下车模显示异常的相关缺陷，相似度得分在0.803~0.820范围内。其中排名第1的BUG-10861687（分屏页面从设置进入导航，1/3屏车模消失，相似度0.820）与当前缺陷高度相关。Top-4命中同类问题，验证了LLM摘要增强检索的有效性。

Stage 4结果与人工分析对比：表3展示了AI与人工分析的对比。AI正确识别了导航模块与Unity渲染服务间的状态交互异常，指出分屏切换中导航主活动生命周期频繁变化为触发条件。人工分析指出具体原因为导航发送isOneTake=false消息，修复方案为分屏下屏蔽对3D发消息的相关接口。AI分析方向正确，能定位到模块级别但未达应用程序编程接口（Application Programming Interface，API）调用级别。责任模块判定与人工一致，推理包括完整7步CoT链路。

表2　Top-4相似缺陷检索结果

排名	BUG ID	相似度	标题	相关性判断
1	10861687	0.820	分屏页面从设置进入导航，1/3屏车模消失	高度相关
2	10925835	0.816	分屏状态下进入导航不显示车模	高度相关
3	10871117	0.810	分屏模式下一镜到底导致车模丢失	高度相关
4	10898830	0.803	分屏进入导航偶现车模位置错误	相关

表3　AI分析结果与人工分析对比

评估维度	AI分析结果	人工分析	对比评价
根因定位	导航模块与Unity渲染服务交互异常；触发条件为分屏切换中导航主活动生命周期频繁变化	导航发送isOneTake=false消息导致车被拉到导航机位	方向正确但精度有限（能定位到模块级交互问题，但未触达具体API调用）
责任模块	导航模块、Unity渲染服务	导航模块（及相关3D交互）	判断一致
修复方案	受限于根因精度，体现为模块级建议	分屏下屏蔽对3D发消息的相关接口	AI暂时无法给出具体代码接口级别的修复建议
推理过程	包括完整的7步CoT链路，且每一步均有日志证据支撑	通常为个人经验总结，无结构化过程	AI具备完整清晰的逻辑和严密的日志证据链

4.3　效率评估

CockpitInsight完整Pipeline耗时约70 s（含视频分析），其中Stage 1a约3 s，Stage 1b约40 s（含VLM逐帧分析与ASR语音识别），Stage 2约9 s，Stage 3约8 s，Stage 4约9 s。相比人工2~4 h，效率提升超99%。

5　讨论

5.1　方法有效性分析

研究表明，四阶段Pipeline各阶段的协同效应是方法有效性的关键。缺陷文本理解阶段的时间点提取与缺陷视频理解阶段的视频屏幕时间戳共同为Stage 2日志过滤提供精确时间锚点，当工单文本缺少精确时间信息时，视频屏幕时间戳可作为关键补充，使三区域过滤在99.6%压缩比下保留关键事件链。Stage 2的精炼日志为Stage 4推理提供核心证据。Stage 3的历史检索使LLM即使缺乏特定业务知识也能给出方向正确的分析。

LLM摘要增强是语义检索质量的关键因素。在未使用LLM摘要的情况下，对原始描述（包含大量模版噪声）直接进行嵌入编码，会导致向量空间中的语

义偏移。LLM摘要将多样化的描述统一压缩为结构化的“现象/原因/关键词”格式，有效消除了噪声和表述差异。

5.2 局限性

本研究存在以下局限性：(1)验证样本量有限，仅完成1个缺陷的深度验证，需扩展至20~30个样本；(2)AI能定位到模块级别，但难以达到API调用级别，需结合代码检索或模型微调。

需要补充说明的是，文中3.3节所提的自适应时间窗口过滤方法依赖人工维护的Tag白名单与关键词模板，跨模块迁移时召回率波动较大；当缺陷涉及未登记的Tag或时间锚点偏移较大时，关键事件可能被误压缩。针对该局限，本文在工程实现中构建一个可与3.3节过滤管线互为替代的LogAgent模块作为后续工作的方向之一：将日志检索能力封装为5个原子工具(总览统计、正则检索、时间窗读取、窗口Tag分布与首次匹配定位)，通过Function Calling协议交由LLM自主迭代调用，通过多轮迭代调用产出日志分析报告，无需为新模块额外维护过滤模板。

当前系统定位为不依赖源码的快速分析管线，大语言模型产生无证据支撑结论(诊断幻觉)的风险客观存在。针对上述局限，后续工作已完成源码联合分析的概念验证：通过引入代码搜索工具，LLM可将日志调用链与模块源码实现进行交叉比对，在代码层面对推理结论加以验证与收敛，预期可大幅降低诊断幻觉。

5.3 推广性

CockpitInsight架构具有良好的可推广性。适配其他车载子系统仅需替换Tag白名单和领域提示词；适配其他整车企业仅需替换缺陷系统字段映射规则。具有结构化工单、大量日志和根因分析需求的场景均可参考本方法。

6 结束语

本文提出了基于大语言模型的四阶段自动分析框架CockpitInsight，融合VLM视频分析、ASR语音识别与LLM文本推理，实现了从缺陷工单到结构化根因报告的全自动生成。

核心贡献包括：(1)自适应时间窗口日志过滤算法，三区域分层策略将17万行日志压缩至约700行(压缩比99.6%)；(2)LLM摘要增强的语义缺陷检索，在1 054条缺陷索引上实现高质量跨项目匹配。

经某整车企业真实数据验证表明，端到端分析耗时约70 s(含视频分析)，相比人工2~4 h提速超99%。生成报告包含完整推理链路、日志证据引用和修复建议，验证了LLM驱动的多模态缺陷分析在智能座舱领域的工程可行性。

未来工作将从4个方向展开：(1)扩大验证规模至20个以上多模块样本并邀请开发人员盲评；(2)构建缺陷-组件-修复知识图谱增强API级定位能力；(3)引入工具调用驱动的LogAgent作为自适应过滤的补充方案，并探索“过滤快速通道+Agent兜底”的混合编排，以提升跨模块场景下的可迁移性；(4)基于已完成概念验证的CodeAgent，引入模块源码作为推理证据，交叉核验日志调用链与代码实现，进一步抑制诊断幻觉。

参 考 文 献

[1] 盖世汽车研究院. 2025年H1中国乘用车智能化市场展望 [EB/OL]. (2025-08-14) [2026-09-09]. https://auto.gasgoo.com/institute/2583.html.

[2] ZHAO W X, ZHOU K, LI J, et al. A Survey of Large Language Models[EB/OL]. (2023-03-31) [2026-09-09]. https://arxiv.org/abs/2303.18223.

[3] LEWIS P, PEREZ E, PIKTUS A, et al. Retrieval-Augmented Generation for Knowledge-intensive NLP Tasks[C]// Advances in Neural Information Processing Systems 33. Red Hook, NY, USA: Curran Associates, Inc., 2020: 9459-9474.

[4] WEI J, WANG X, SCHUURMANS D, et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models[C]// Advances in Neural Information Processing Systems 35. New Orleans, LA, USA: Neural Information Processing Systems Foundation, Inc., 2022: 24824-24837.

[5] DU M, LI F, ZHENG G, et al. DeepLog: Anomaly Detection and Diagnosis From System Logs Through Deep Learning[C]// Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas, TX, USA: ACM, 2017: 1285-1298.

[6] MENG W, LIU Y, ZHU Y, et al. LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs[C]// Proceedings of the 28th International Joint Conference on Artificial Intelligence. Macau, China: AAAI Press, 2019: 4739-4745.

[7] HE P, ZHU J, ZHENG Z, et al. Drain: An Online Log Parsing Approach with Fixed Depth Tree[C]// 2017 IEEE International Conference on Web Services (ICWS). Honolulu, HI, USA: IEEE, 2017: 33-40.

[8] ZHI C, CHENG L, LIU M, et al. LLM-Powered Zero-Shot

Online Log Parsing[C]// 2024 IEEE International Conference on Web Services (ICWS). Shenzhen, China: IEEE, 2024: 877-887.

[9] LAMKANFI A, DEMEYER S, GIGER E, et al. Predicting the Severity of A Reported Bug[C]// 2010 7th IEEE Working Conference on Mining Software Repositories (MSR). Cape Town, South Africa: IEEE, 2010: 1-10.

[10] NGUYEN A T, NGUYEN T T, NGUYEN T N, et al. Duplicate Bug Report Detection with A Combination of Information Retrieval and Topic Modeling[C]// Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE). Essen, Germany: ACM, 2012: 70-79.

[11] XIA C S, ZHANG L. Keep the Conversation Going: Fixing 162 Out of 337 Bugs for $0.42 Each Using ChatGPT[C]// Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2024). New York: ACM, 2024: 819-831.

[12] ZHAO Y, LIU Z, ZHENG Y, LAM K Y. Attribution Techniques for Mitigating Hallucinated Information in RAG Systems: A Survey[C]// 2026 8th International Conference on Artificial Intelligence in Information and Communication (ICAIIC). Tokyo, Japan: IEEE, 2026: 687-692.

[13] ELCHAFEI P, SWAIN M, MASOUDIAN S, et al. Facet-level Tracing of Evidence Uncertainty and Hallucination in RAG[EB/OL]. (2026-04-10) [2026-09-09]. https://arxiv.org/abs/2604.09174.

[14] ROY D, ZHANG X, BHAVE R, BANSAL C, LAS-CASAS P, FONSECA R, et al. Exploring LLM-based Agents for Root Cause Analysis[EB/OL]. (2024-03-07) [2026-09-09]. https://arxiv.org/abs/2403.04123.

[15] International Organization for Standardization. ISO 26262-6:2018 Road Vehicles-Functional Safety-Part 6: Product Development at the Software Level[S]. Geneva: ISO, 2018.

[16] VDA QMC Working Group 13, Automotive SIG. Automotive SPICE Process Assessment/Reference Model v3.1[S]. Berlin: VDA QMC, 2017.

（责任编辑　**梵玲**）

《汽车工程师》征稿启事

《汽车工程师》于1974年创刊，是由中国第一汽车集团有限公司主管、中国第一汽车股份有限公司主办的国内外公开发行的汽车应用技术类学术期刊，为RCCSE中国准核心学术期刊、CACJ中国应用型扩展期刊，被哥白尼精选数据库、EBSCO学术数据库、欧洲学术出版中心（EuroPub）数据库、J-Gate数据库收录。

《汽车工程师》以报道汽车产品设计领域的技术研究成果为主，专注于技术开发与应用层面的创新成果，提供汽车设计创新解决方案。《汽车工程师》将把握电动化、智能化、网联化、共享化的汽车技术主流发展趋势，努力吸收优质稿源，为我国汽车工程技术创新能力提升贡献力量。重点征稿方向包括：锂离子电池与新体系电池、动力电池热管理与热失控、燃料电池及其关键零部件、车用电机及其控制器、自动驾驶汽车环境感知与感知融合、自动驾驶汽车定位与建图、自动驾驶汽车测试、人工驾驶-无人驾驶车辆混行条件下的自动驾驶控制、智能线控底盘、一体化底盘、功能安全和预期功能安全。

热忱欢迎汽车行业的专家学者不吝赐稿，反映国家重点扶持项目、自然科学基金项目和其他重点项目等研究成果的稿件将优先发表，我们期待与您共同践行“把论文写在祖国大地上”的指示精神，为强大中国汽车工业作贡献！

本刊使用协同采编系统处理稿件，请登陆期刊网站注册后投稿，投稿要求见本刊网站中“下载中心”栏目的作者指南，网址：http://tjqc.cbpt.cnki.net。

《汽车工程师》编辑部