
# DTS311TC FINAL YEAR PROJECT

## *Player-Aware Intelligent Monitoring and Operations Navigator*

## Proposal Report

In Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Engineering


| | | |
|---|---|---|
| Student Name | : | Taimingwang Liu |
| Student ID | : | 2037690 |
| Supervisor | : | Xihan Bian |

School of AI and Advanced Computing
Xi'an Jiaotong-Liverpool University

November 2025

# Abstract

Apply the font of Times New Roman to the paragraphs of the abstract using font size of 12. An abstract is usually one to three paragraphs long with a length of 150 to 350 words.

# Contents

# 1 Introduction

## 1.1 Problem Setting & Motivation

近年来，面向玩家的智能交互快速涌现：从*AI*游戏主播/虚拟角色到*LLM*驱动的*NPC*/插件，社区与产业案例显示"大模型×游戏交互"具备关注度与潜在影响（game changer potential）。然而，这些实践多依赖场景定制，缺乏统一动作接口与可复现实验协议；相应研究正在通过统一评测/模块消融与污染控制/协议一致等方法学加以弥合[1], [2]。本文讨论伴随式（*companion-style*）实时助手，并将相关问题置于仅基于*GUI*（*GCC*）与低延迟体验的工作设定下加以界定。

**Industry/Community Signals** 在社区层面，*Neuro-sama* 的*AI* 游戏主播（*AI streamer*）现象展示了大模型驱动的持续互动与情绪共鸣能力[3]；同时，叙事解谜作品*AI2U: With You 'Til The End* 体现了"对话即操作（*dialogue-as-action*）"与高交互度（LLM-controlled NPCs）的设计潜力[4]。围绕该方向的开源复现与二次开发——如*Open LLM VTuber*、*Airi Project*、*Kimjammer-Neuro* 等——持续出现，反映出应用生态的活跃[5]–[7]。上述社区与产业案例仅作为动机与生态线索，并不作为本文方法有效性的学术证据。

作为市场信号，AI VTuber（以*Neuro-sama* 为代表）已在主流平台获得大规模关注：其在2025 年1 月创造Twitch *Hype Train* 世界纪录（Level 111，~85K 付费订阅、~1.2M bits），相关媒体与行业统计均有报道；其频道长期维持数十万粉丝规模与高并发活跃[8]–[11]。同时，直播总体观看时长处于高位（如Twitch 2024 年全年约18.5–20.8 B 小时；2024 Q4 全行业约21 B 小时），显示"AI×直播/游戏"具备现实受众与可观市场体量[12], [13].

## 1.2 Scope & Working Definitions

本文将动作接口（action interface）的工作定义限定为**General Computer Control (GCC)** 的*GUI*范式：*screen-in, keyboard/mouse-out* 的人类同态接口（human-homomorphic interface）。代表性工作*CRADLE* 报告了在不依赖应用API 的前提下完成长链路桌面/游戏任务的可行性与系统结构（规划/技能整理/反思/记忆）[14]。此外，*Model Context Protocol (MCP)* 可作为内部模块/技能编排（registration/orchestration）的通用思路，与具体的输出通道无直接绑定；"统一评测/消融"与*plug-and-play* 思路可见*ORAK*[1]；基于*procedural generation* 的OOD 方法学可见*Benchmarking-VLA-VLM*[15]；将真实游戏"转化为可靠评测"的协议化实践可见*lmgame*[2]。在*GUI* 场景中，*UI-Venus* 强调纯截图（*screenshot-only*）输入与结构化动作（*structured output*）的端到端导航[16]；而*V-MAGE* 聚焦*visual-only/continuous-space* 的视觉中心评测[17]。上述工作将在第二部分的相关研究中系统梳理。

**Screenshot-only navigation (GUI).** 文献显示，在真实平台上，纯截图输入+ 结构化动作输出亦可实现端到端导航并取得具有竞争力的结果（如AndroidWorld 的*pass@1* 与ScreenSpot 系列的屏幕定位任务）[16]。

作为多模态大模型（MLLM）的通用结构示意，图 1 展示了文本经*tokenizer* 输入LLM、非文本模态经*Multimodal Encoder/Projector* 对齐后与文本融合的典型流程[18]。

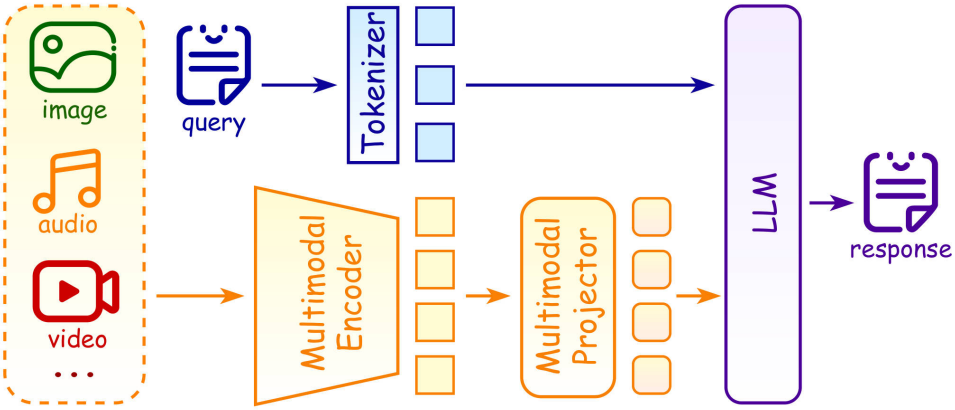


Figure 1: The overall architecture of MLLMs [18].

作为多模态交互智能体（Agent AI）的概念性综述，作者从“下一步具身动作预测（next-embodied action prediction）”出发，讨论外部知识、人类反馈与多传感输入在*grounded* 场景下提升稳健性的作用[19]。面向GUI 自动化的综述则系统梳理了以LLM 为“中枢”的GUI 智能体在框架、训练数据/大动作模型（*LAM*）、评测基准与指标方面的进展与挑战[20]；另有面向(M)LLM-based GUI agents 的综述从感知—探索/知识—规划—交互四组成进行框架化对齐，并指出评测方法学与标准化的挑战[21]；在更上层的OS 范畴，*OS Agents* 综述提出“环境/观测/动作—理解/规划/落地”的要素与能力图谱[22]。

## 1.3 Key Challenges

（i）长链路稳定性（**long-horizon stability**）：在*GUI*（*GCC*）通道上，错误累积与状态漂移更易放大；文献通过*skills*/反思/记忆等管线缓解，但挑战仍存[14]；（ii）视觉中心定位与记忆（**vision-centric grounding & memory**）：*visual-only/continuous-space* 设定对定位/时机/视觉记忆/高层推理提出更高要求[17]；（iii）**OOD**与协议一致性（**OOD & protocol consistency**）：评测需在过程生成与变量可控的条件下比较架构/数据/后处理，减少不可比性[15]；（iv）提示方差与污染（**prompt variance & contamination**）：将“游戏→评测”落到可复现协议需稳定交互回路并记录后处理[2]；（v）无效动作与幻觉（**invalid actions & hallucination**）：结构化输出/约束解码可降低*invalid action*，但*think–action mismatch* 等现象仍被报告[16]；（vi）时延与交互体验（**latency & UX**）：实时伴随式场景强调语音往返（voice RTT）与帧到提示的响应时间，需与稳定性指标共同考量[1], [2]。

## 1.4 Our Positioning & Contributions

**TODO:** (1) 研究定位：GUI（GCC）下的伴随式助手设定；(2) 概念性模块化：以MCP-style 作为内部“技能/工具总线”进行注册/编排（与输出通道无关）；(3) 评测要素：统一的任务脚本与指标（advice adoption, voice RTT, macro success 等）；(4) 预期交付物：原型/评测脚本/文档（按C.1中的Expected Results表述）。

## 1.5 Design Principles & System Preview

**TODO:** 系统流一句话：*screen/audio* → *VLM* → *LLM/agentic*（planning/memory/reflection）→ *MCP-style*（技能注册/路由）→ *GUI*执行（kb/mouse）→ *safety*（permissions, rollback, kill-switch）。设计原则：结构化输出（structured output）、可审计（auditability）、可复现（reproducibility）。

# 2 Literature Review

## 2.1 Perception: Modalities & Grounding

该方向通常以屏幕帧（screen frames）或短视频（video clips）为主输入，辅以窗口/坐标等轻量上下文；可选接入音频（audio）以形成语音闭环（ASR/TTS）。多模态模型承担检测/描述/定位（detection/description/grounding）、UI 元素识别与状态读出等能力；相较直接产出动作的*VLA*，*VLM+tool* 将视觉理解与动作执行解耦，通过结构化调用（*structured tool calls*）或技能库（*skills/macros*）闭环。代表实践包括：*UI-Venus* 在*screenshot-only* 条件下以结构化动作实现端到端导航[16]；*V-MAGE* 强调*visual-only/continuous-space* 设定下的定位、时机与视觉记忆压力[17]；*lmgame* 提供感知/记忆脚手架（scaffolds）以稳定交互与提示方差[2]。上述脉络共同勾勒了输入模态与*grounding* 能力的术语与范围。[2], [16], [17]

**Agent AI**（**multimodal interaction survey**） 该综述以“*Agent AI*”为工作定义：感知多模态输入（视觉/语言/环境信号）并在具身或虚拟环境中产生动作（embodied actions）的交互系统；作者从“下一步具身动作预测（next-embodied action prediction）”出发，讨论外部知识（external knowledge）、多传感输入（multi-sensory inputs）与人类反馈（human feedback）在*grounded* 场景下的作用，并建议以虚拟/模拟环境加速研究进展[19]。该视角为“感知—定位—交互”的表述提供了概念锚点。

## 2.2 Action Interfaces: GUI (GCC) & MCP-style Orchestration

动作接口层面，**GUI** 路线以**General Computer Control (GCC)** 为统一通道（*screen-in, keyboard/mouse-out*），强调跨应用/跨游戏的可迁移性（portability）与人类同态交互（human-homomorphic interface）。代表性工作*CRADLE* 展示：在不依赖应用专用接口的前提下，结合规划—技能整理（*skill curation/registry*）—反思—记忆的管线亦可完成长链路任务（desktop/games）[14]；与此同时，**MCP**（Model Context Protocol）提供模块注册/编排（*module registration/orchestration*）的协议化思路，使*skills/macros*、*planning*、*memory*、*reflection* 能在统一接口下组织与对比[1]。由此形成“以GUI 为执行通道、以协议化编排（MCP-style）组织模块”的常见范式。

如Figure 2 所示，CRADLE 以统一的GUI（GCC）通道展示了从“屏幕输入→内在推理→键鼠控制”的闭环。

**UI-Venus**（**screenshot-only**） 端到端GUI 导航，无需planner/A11y；截图→ 结构化动作 在真实平台报告了具有竞争力的结果（如AndroidWorld 的pass@1 与ScreenSpot 系列定位）[16]，强化了“GUI 统一通道”的可达性认识。

**LLM-brained GUI agents**（**survey**） 该综述以“LLM 为中枢的GUI 智能体”为统一对象，围绕GUI 自动化的框架、数据与训练、面向动作的专门化模型（*large action*

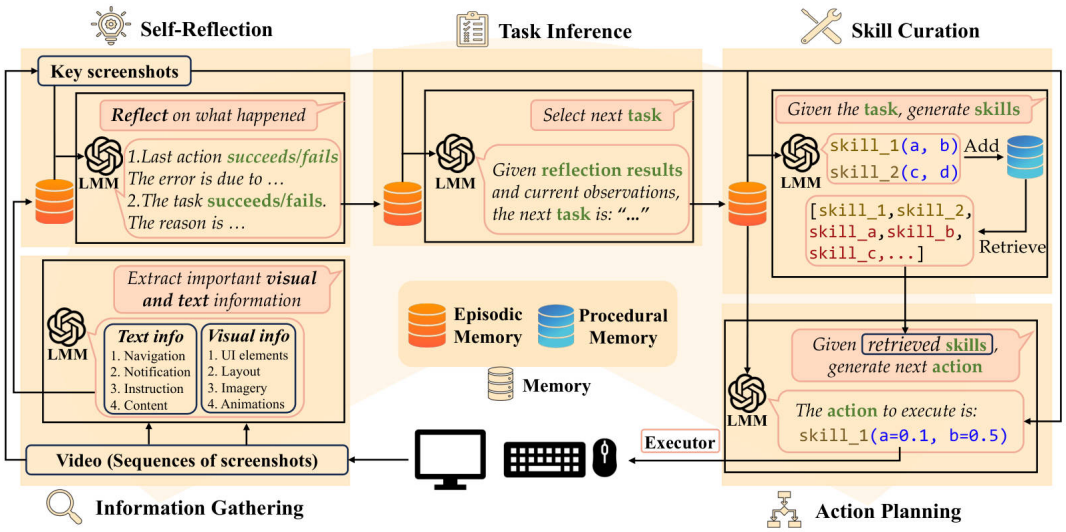


Figure 2: An overview of the CRADLE framework: CRADLE takes video from the computer screen as input and outputs computer keyboard and mouse control determined through inner reasoning (planning, skill curation, reflection, memory) [14].

*models, LAM*）与评测基准/指标展开，总结跨Web/移动/桌面平台的通用交互能力与挑战，并提出若干路线图[20]。

**(M)LLM-based GUI agents（survey）** 该综述将GUI 智能体拆解为四大组件：*perception*（多模态理解）、*exploration/knowledge*（内部模型、历史回放与外部检索）、*planning*（任务分解与推理）与*interaction*（动作生成与安全控制），并回顾跨桌面/移动/Web 的研究进展；作者指出元素定位、知识检索、长时规划与安全执行控制仍是挑战，同时强调评测在指标与协议上的标准化需求[21]。两篇综述共同提供了动作接口与系统组成的对齐视角。

**OS Agents（OS-level scope survey）** 在OS 视角下，(M)LLM 驱动的计算设备智能体通过操作系统提供的接口（如GUI/CLI）跨电脑/手机/浏览器执行任务；综述提出"环境—观测空间—动作空间"的要素划分，并将"理解/规划/动作落地（grounding）"作为核心能力，系统回顾基础模型、代理框架、评测与安全威胁[22]。该范围化视角有助于统一"设备—接口—能力"的讨论语言。

## 2.3 Agentic Modules: Planning, Memory, Reflection, Skills

常见*agentic modules* 包括：*planning*（分解与策略选择）、*memory*（短长时与用户偏好）、*self-reflection*（纠错与风格一致）与*skills/macros*（原子→复合）。例如，*CRADLE* 组合*planning/skills/reflection/memory* 以缓解长链路误差累积；*ORAK* 在统一评测中对上述模块进行消融比较[1], [14]；*UI-Venus* 在训练与数据层面探索轨迹历史对齐与稀疏动作增强**uivenus_rft**。该类机制为长链路稳定性与一致性提供了可分析的结构要素。

历史对齐与稀疏动作增强 *Self-Evolving Trajectory History Alignment & Sparse Action Enhancement*：用当前模型重写历史“思维—动作”轨迹以对齐风格/细节，并上采样稀疏但关键动作（如LongPress），报告了对长链路一致性与泛化的改善**uivenus_rft**。此外，有工作将“下一步具身动作预测（next-embodied action prediction）”与人类反馈并置为提升*agentic* 能力的关键因素，强调在*grounded* 环境中校准策略与记忆的重要性[19]。

## 2.4 Learning Paradigms: Zero-shot, RAG, Finetune, IL/RL, Distillation

在学习与推理范式上，零样本/提示工程（prompting）、检索增强（RAG）、轻量微调（LoRA）、模仿/强化（IL/RL）与蒸馏（distillation）并行发展。以“指令化（instructionalization）”增强RL 代理的上下文理解是一条代表性路线：*R2-Play* 将多模态游戏指令（MGI）并入*Decision Transformer*（*DTGI*），并通过超网络（*SHyperGenerator*）在训练任务与未见任务间共享知识；作者报告多模态指令较文本/轨迹单模态在多任务与泛化上更优（动机见Figure 3），且MGI 的三段式结构——*game description*、*game trajectory*、*game guidance*（含动作、语言引导及关键元素位置）——给出清晰的指令模板（见Figure 4）[23]。

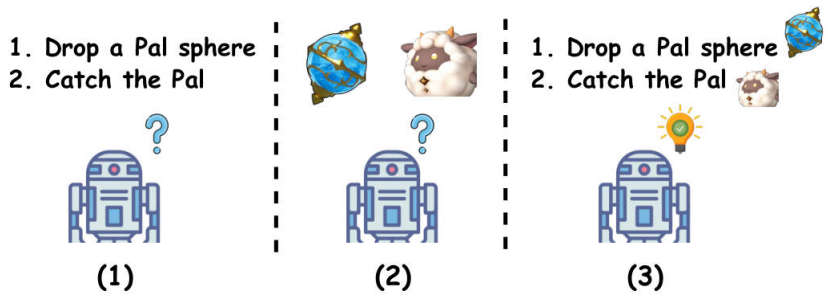


Figure 3: Imagine an agent learning to play Palworld. (1) The agent exhibits confusion when only relying on textual guidance. (2) The agent is confused when presented with images of a Pal sphere and a Pal. (3) The agent understands how to catch a pet through *multimodal guidance*, which combines textual guidance with images of the Pal sphere and Pal [23].

**RFT（GRPO）与动作粒度奖励** 将奖励拆分为格式/动作类型/坐标/内容四部分并加权，以同时度量结构化输出合规性与细粒度定位/文本输入正确性，作为GUI 导航中RL-finetune 的代表做法之一**uivenus_rft**。

另一方面，“工具增强型MLLM（tool-augmented MLLM）”的综述从“数据—任务—评测”三条主线梳理外部工具（API、专家模型、知识库等）的作用边界：任务侧涵盖多模态RAG、推理、幻觉、安全、代理与视频感知，流程侧以MRAG 的“检索—重排—整合”三段式为例，评测侧指出既有指标难以全面刻画多模态生成与对齐[18]。这些观察为讨论学习范式与评测要素提供了方法学参考。


Figure 4: An illustrative example of *multimodal game instructions (MGI)*. Each instruction consists of three sections: *game description*, *game trajectory*, and *game guidance* (including action, language guidance, and the position of key elements) [23].

## 2.5 Benchmarks & Datasets (OS-like, Games, Desktop)

**ORAK**（统一评测/消融/MCP 思想） *ORAK* 通过MCP 实现*plug-and-play* 的代理—环境解耦，并在统一配置下检验*planning / reflection / memory / skills* 等*agentic modules* 的边际贡献（ablation），配套*Leaderboard/Battle Arena* 与训练轨迹数据（fine-tuning trajectories），将机制—性能—配置一体化呈现[1]。该体例凸显了“统一评测—模块消融”的可比性价值。

**Procedural-generation**（**OOD** 方法学） 以*procedural generation* 构造OOD 与多步任务压力，统一比较*VLA/VLM* 在架构/训练数据/输出后处理等变量下的稳健性，并配套工具链保证*reproducibility*[15]。这是“变量可控—开放式难度”方向的典型方法学线索。

**lmgame-Bench**（脚手架与污染控制） *lmgame* 将“游戏→评测”系统化：用*Gym-style* 接口与*perception/memory scaffolds* 稳定prompt 并剔除“污染”，在多模型下获得良好分离度，并通过相关性分析展示不同游戏探测的能力混合；另报告单一游戏的RL 训练对未见游戏/外部规划任务存在迁移[2]。整体上，评测组织正从“单点游戏”走向“脚手架化、协议一致”的可复现比较。

...为减少提示方差并抑制污染，lmgame 以模块化脚手架稳定“感知—记忆—推理”的交互回路（见Figure 5）。

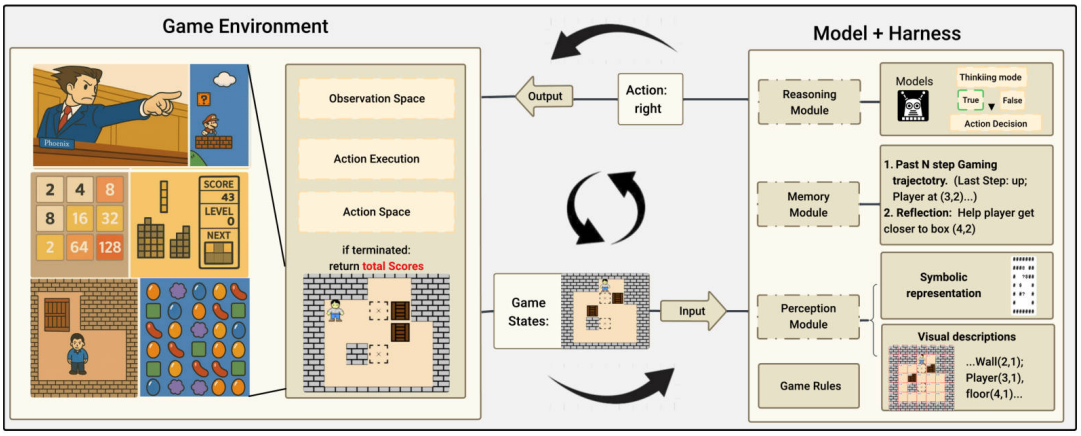


Figure 5: lmgame-Bench uses modular harnesses—such as perception, memory, and reasoning modules—to systematically extend a model's game-playing capabilities, enabling iterative interaction loops with a simulated game environment [2].

**V-MAGE**（**vision-centric, visual-only, continuous-space**） 该框架以仅视觉输入与连续空间的游戏环境评测多模态模型的视觉中心能力，覆盖定位、轨迹追踪、时机、视觉记忆及更高层时序推理；其评测管线支持分离“模型/策略”，并采用Elo 风格排名进行相对强度比较；作者报告模型与人类表现存在差距、常见感知错误与锚定偏差，且有限历史上下文会限制长时规划[17]。这一视觉中心脉络补充了跨类型真实游戏基准的视角。

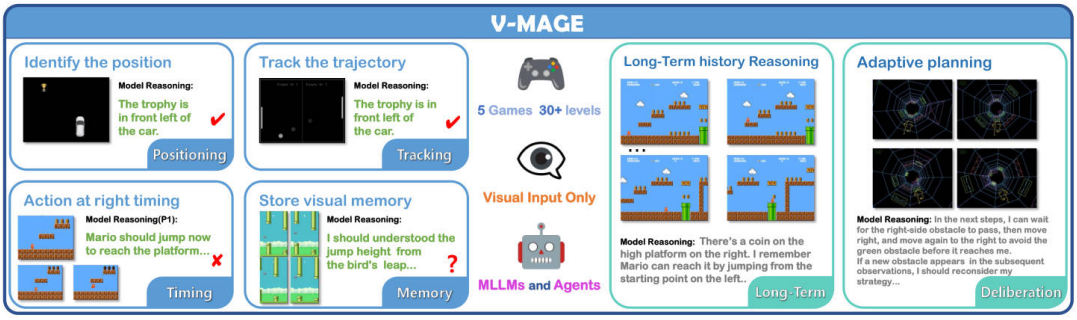


Figure 6: The overview of the V-MAGE benchmark, designed to evaluate vision-centric capabilities and higher-level reasoning of MLLMs across 5 free-form games with 30+ levels [17].

## 2.6 Evaluation Protocols & Metrics

常见客观指标：任务成功率（*success/pass@k*）、完成时间（*time-to-completion*）、误操作/回滚（*misclick/rollback rate*）与延迟（*latency*，如语音往返*voice RTT*）；主观/行为指标：采纳程度（*advice adoption*）与满意度等[1], [2], [15]。在类别分布不均（class imbalance）情形，*macro-averaged* 与*micro* 指标并举以减轻多数类偏置[15]。协议层面，文献强调结构化输出与约束解码（*structured output & constrained decoding*）以降低无效动作与坐标偏差，也有将无效动作率纳入评估与训练信号设计（动作拆分为格式/类型/坐标/内容四粒度）的做法[16]；“游戏→可靠评测”路线强调记录后处理/解码策略（*post-processing*）、提示方差与污染控制，以减少实现细节对可比性的影响[2]。跨任务比较上，*Elo* 风格排名（*Elo-style ranking*）可缓解关卡难度不均与多任务汇总问题[17]；而*procedural generation* 的方法学将架构/数据/后处理视作可控变量进行统一对比，强调OOD 稳健性与可复现性[15]。以上做法为后续指标与协议的组织提供了参考图谱。

## 2.7 Deployment & Real-time Considerations

部署相关研究聚焦资源与实时性约束：本地—云混合（*local–cloud hybrid*）与推理量化（*inference quantization*，*INT4/FP8*）降低时延与成本；流式解码（*streaming decoding*）与语音中断（*barge-in*）缩短交互回路；并评估对帧率与CPU/GPU 占用的影响。为保证可比性，协议侧采用固定提示、记录*post-processing* 与环境版本；在真实设备/平台的在线评测亦逐步增多（如AndroidWorld 与脚手架化交互）[2], [16]。这些观察勾勒出“实时—资源—协议”三方面的工程边界。

## 2.8 Safety, Permissions & Robustness

文献强调权限模型（*permission scoping/whitelisting*）与操作确认（*confirmation*）以约束高风险动作；影子模式（*shadow execution*）先预测后执行以降低副作用，并配套回滚（*rollback*）与急停（*kill-switch*）保障可逆性与故障恢复。在GUI 场景

中，*think–action mismatch* 揭示了多模态模型可能产生的“幻觉（hallucination）”与不一致风险，提示需要日志与审计（*auditability*）支持溯源与复查[16]。总体上，权限边界、影子执行与可审计性构成了“安全—鲁棒”的基本支架。

如Figure 7 所示，think–action 不一致（mismatch）揭示了MLLM 的“幻觉”（hallucination）风险[16]。

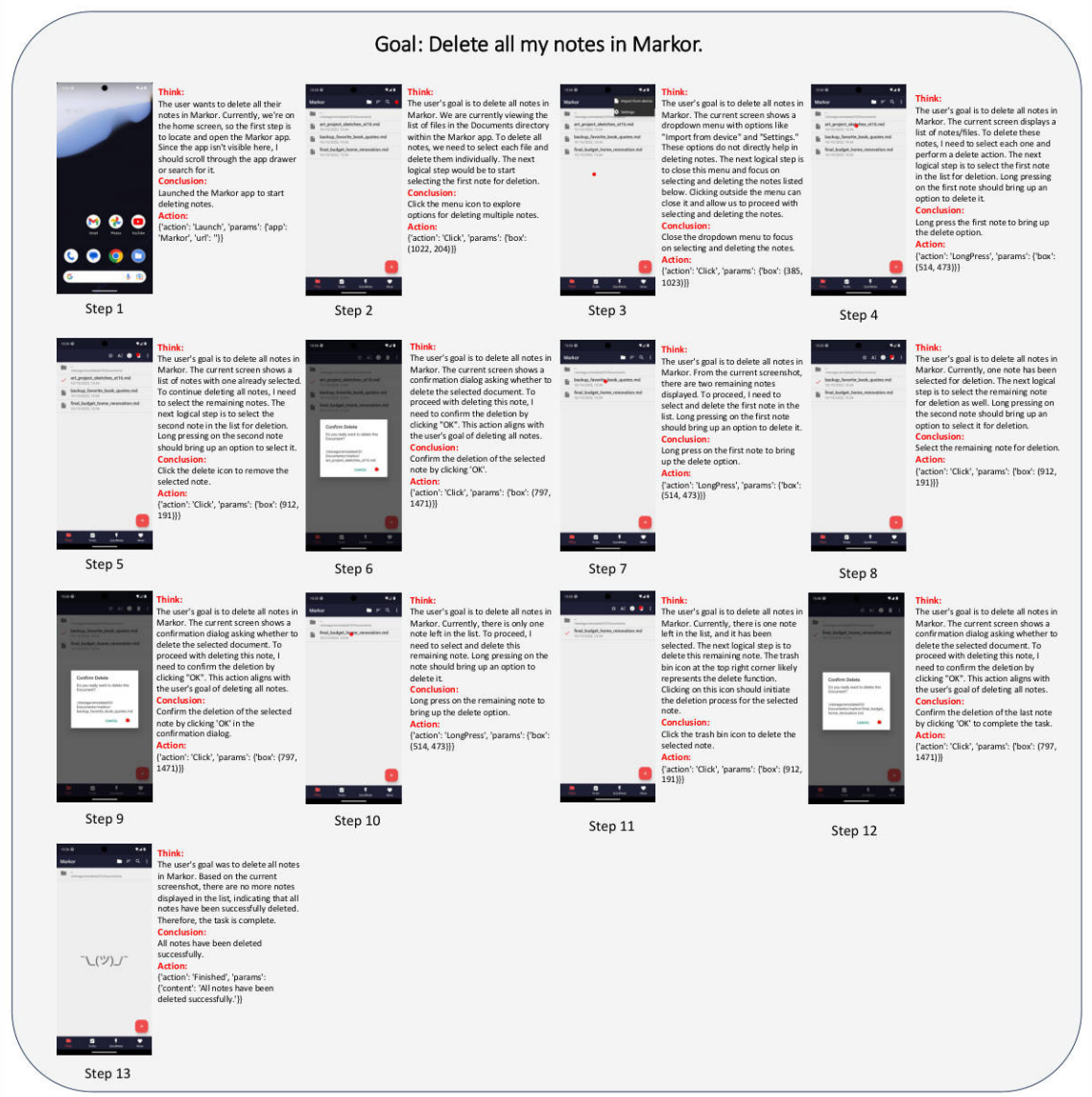


Figure 7: One trace of UI-Venus on the task named MarkorDeleteAllNotes in AndroidWorld. We can observe that UI-Venus successfully achieves the goal and has the reflection ability in Step 3. However, there also exists the conflict between think and action in Step 5, remaining as a future work about how to solve MLLM's hallucination. [16]

## 2.9 Synthesis: Trends, Gaps & Our Niche

总体趋势是：在GUI（GCC）通道上引入协议化/模块化编排以支撑可复现实验与消融；跨类型真实游戏的统一评测（如*ORAK*）与*visual-only/continuous-space* 的视觉中

心评测（如*V-MAGE*）并行发展；将脚手架/污染控制纳入协议（如*lmgame-Bench*）成为共识[1], [2], [17]。同时，沿着*grounded* 的感知—行动—人类反馈闭环，研究正逐步把注意力从单点Demo 推向过程变量与稳健性[19]。

# 3 Project Plan

## 3.1 Proposed Solution / Methodology

## 3.2 Experimental Design

## 3.3 Expected Results

## 3.4 Progress Analysis and Gantt Chart

### 3.4.1 Risk & Ethics

# 4 Conclusion

# References


[1] D. Park *et al.*, "Orak: A foundational benchmark for training and evaluating llm agents on diverse video games," 2025, arXiv:2506.03610. arXiv: 2506.03610.

[2] L. Hu *et al.*, "Lmgame-bench: How good are llms at playing games?, 2025a," *URL https://arxiv. org/abs/2505.15146*,

[3] Vedal and Neuro-sama, *Neuro-sama official youtube channel*, https://www.youtube.com/@Neurosama, Accessed: 2025-10-10, 2022.

[4] AlterStaff, *Ai2u: With you 'til the end*, https://store.steampowered.com/app/2880730/AI2U_With_You_Til_The_End/, Accessed: 2025-10-10, 2025.

[5] O.-L.-V. contributors, *Open-llm-vtuber: An open-source ai vtuber framework*, https://github.com/Open-LLM-VTuber/Open-LLM-VTuber, Accessed: 2025-10-10, 2025.

[6] moeru-ai, *Airi: Ai waifu / virtual character container inspired by neuro-sama*, https://github.com/moeru-ai/airi, Accessed: 2025-10-10, 2025.

[7] kimjammer, *Neuro: A local-model recreation of neuro-sama*, https://github.com/kimjammer/Neuro, Accessed: 2025-10-10, 2025.

[8] "Vedal's ai vtuber neuro-sama sets new twitch hype train world record." Accessed: 2025-10-11, Streams Charts. [Online]. Available: https://streamscharts.com/news/vedals-ai-vtuber-neuro-twitch-hype-train-record.

[9] P. Staff. "With valorant's help, this ai vtuber just beat a massive twitch record." Accessed: 2025-10-11, PCGamesN. [Online]. Available: https://www.pcgamesn.com/valorant/neuro-sama-twitch-record.

[10] "Vedal987 — streamer overview & stats." Accessed: 2025-10-11, TwitchTracker. [Online]. Available: https://twitchtracker.com/vedal987.

[11] C. Xiang. "Meet neuro-sama, the ai twitch streamer who plays minecraft, sings karaoke, loves art." Accessed: 2025-10-11, Bloomberg. [Online]. Available: https://www.bloomberg.com/news/newsletters/2023-06-16/neuro-sama-an-ai-twitch-influencer-plays-minecraft-sings-karaoke-loves-art.

[12] StreamElements. "State of the stream: Twitch 2024 year in review." Accessed: 2025-10-11. [Online]. Available: https://blog.streamelements.com/state-of-the-stream-twitch-2024-year-in-review-ef4d739e9be9.

[13] "Q4 2024 global live streaming landscape." Accessed: 2025-10-11, Streams Charts. [Online]. Available: https://streamscharts.com/news/q4-2024-global-livestreaming-landscape.

[14] W. Tan *et al.*, "Cradle: Empowering foundation agents towards general computer control," *arXiv preprint arXiv:2403.03186*, 2024.

[15] P. Guruprasad, Y. Wang, S. Chowdhury, H. Sikka, and P. P. Liang, "Benchmarking vision, language, & action models in procedurally generated, open ended action environments," *arXiv preprint arXiv:2505.05540*, 2025.

[16] Z. Gu *et al.*, "Ui-venus technical report: Building high-performance ui agents with rft," *arXiv preprint arXiv:2508.10833*, 2025.

[17] X. Zheng *et al.*, "V-mage: A game evaluation framework for assessing vision-centric capabilities in multimodal large language models," *arXiv preprint arXiv:2504.06148*, 2025.

[18] W. An, J. Nie, Y. Wu, F. Tian, S. Lu, and Q. Zheng, "Empowering multimodal llms with external tools: A comprehensive survey," *arXiv preprint arXiv:2508.10955*, 2025.

[19] Z. Durante *et al.*, "Agent ai: Surveying the horizons of multimodal interaction," *arXiv preprint arXiv:2401.03568*, 2024.

[20] C. Zhang *et al.*, "Large language model-brained gui agents: A survey," *arXiv preprint arXiv:2411.18279*, 2024.

[21] F. Tang *et al.*, "A survey on (m) llm-based gui agents," *arXiv preprint arXiv:2504.13865*, 2025.

[22] X. Hu *et al.*, *Os agents: A survey on mllm-based agents for computer, phone and browser use*, 2024.

[23] Y. Jin *et al.*, "Read to play (r2-play): Decision transformer with multimodal game instruction," *arXiv preprint arXiv:2402.04154*, 2024.

# Appendix A. Title of Appendix A

## A.1 Appendix Heading 1

Text of the appendix goes here

## A.2 Appendix Heading 2

Text of the appendix goes here

## A.3 Appendix Table and Figure Captions

In appendices, table and figure caption labels and numbers are typed in manually (e.g., Table A1, Table A2, etc.). These do not get generated into the lists that appear after the Table of Contents.

# Appendix B. Title of Appendix B

Text of the appendix goes here if there is only a single heading.