XaiR：整合大型语言模型与物理世界的 XR 平台

斯鲁蒂-斯里尼迪*
卡内基梅隆大学

爱德华-卢*
卡内基梅隆大学

Anthony Rowe* 卡内基
梅隆大学博世研究中心

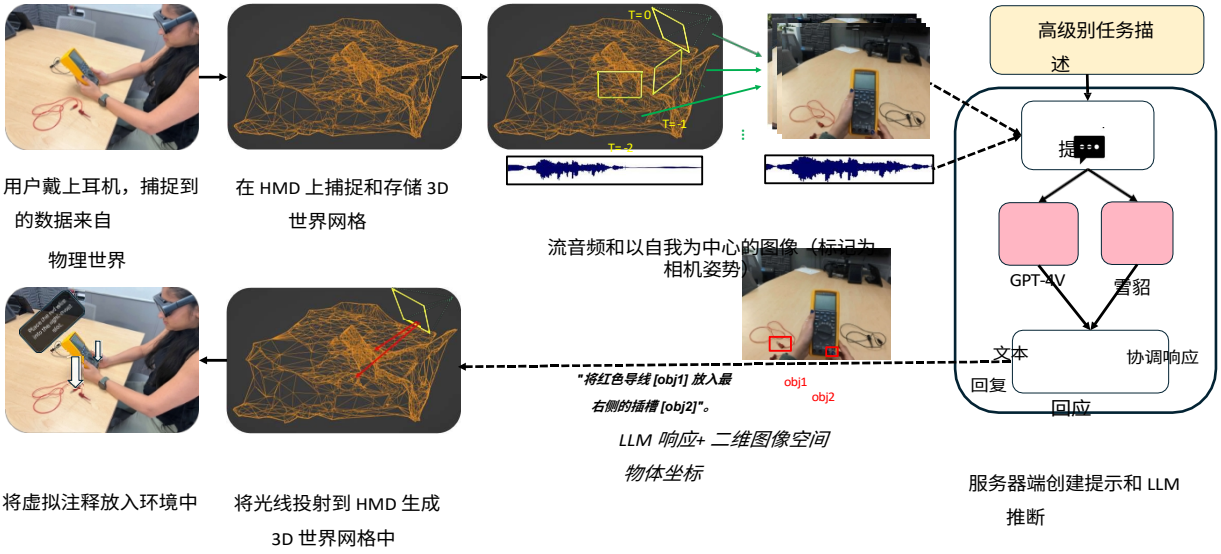


图 1： XaiR 数据流概览：该图说明了如何使用多模态大语言模型来自动放置和生成 XR 内容。它展示了从捕捉 HMD 上的自我中心输入到在物理环境中放置精确的 AR 内容的过程，突出了认知助手在实时任务指导中的作用。

摘要

本文讨论了多模态大语言模型（MLLMs）与扩展现实（XR）头戴式设备的整合，重点是增强机器对物理空间的理解。通过将多模态大语言模型的语境能力与 XR 的感官输入相结合，有可能实现更直观的空间交互。然而，由于 MLLMs 在处理 3D 输入方面存在固有的局限性，而且对 XR 头显的资源需求很大，因此这种整合面临着挑战。我们介绍了 XaiR，这是一个促进 MLLM 与 XR 应用程序集成的平台。XaiR 采用分离式架构，将复杂的 MLLM 操作卸载到服务器上，同时在耳机上处理 3D 世界。这种设置可以管理多种输入模式和并行模型，并将它们与实时姿势数据联系起来，从而改进物理场景中的 AR 内容放置。我们用一款 "认知助手 "应用测试了 XaiR 的有效性，该应用可指导用户完成煮咖啡或组装家具等任务。15 人参与的研究结果表明，任务引导的准确率超过 90%，AR 内容锚定的准确率达到 85%。此外，我们还将 MLLM 与认知助手任务中的人类操作员进行了对比评估，从而深入了解了捕获数据的质量以及认知助手任务性能的当前差距。

1 导言

长期以来，扩展现实（XR）与人工智能（AI）的融合一直是 XR 爱好者的梦想。最近，多模态大语言模型（MLLM）和 XR 头戴式耳机技术的发展使这两种技术的整合成为可能。

*e-mail：{ssrinidh, elu2, agr}@andrew.cmu.edu

域越来越可行。在本文中，我们将探讨几个关键问题：XR 头显如何向 MLLMs 传输以自我为中心的物理信息？MLLMs 能否运用类似人类的感官来解释复杂的空间场景？我们通过这些技术提供功能性认知辅助的效果如何？如果成功，这些类型的系统将彻底改变用于 XR 系统编程的抽象水平，并释放出大幅提高人类效率的潜力。

为了探讨这些问题，我们介绍了 *XaiR*[1]，这是一个在 MLLM 推理和真实世界环境背景之间架起桥梁的平台。高层工作流程如图 1 所示，以传感器数据为输入，经过各种 LLM 处理步骤，生成与场景中物体位置相关的 AR 内容。与头戴式设备（HMD）上主要处理口头询问的典型 LLM 应用程序不同[1, 7]，XaiR 整合了实时 3D 世界网格，以增强情景语境并支持 AR 注释，使用户的交互行为更加身临其境，超越了简单的问答任务。三维世界数据对于许多任务来说都很重要，因为在这些任务中，组件可能不在场景中的当前视图范围内。例如，试想一个系统需要训练用户完成一项任务，而这项任务需要的对象存储在远离用户视角的地方。

目前，LLM 主要针对互联网上的文本和图像数据进行训练，因此很难整合 XR 应用所需的关键 3D 空间数据。这就意味着，虽然一些 LLM 可以处理 2D 图像和视频[16, 17, 25, 32, 34, 36]，但它们处理 3D 数据的能力仍不确定。一种可能的解决方法是直接在 3D 数据集上训练模型，尽管获取必要的大规模数据本身就是一种挑战。XR 头显受限于有限的内存和处理能力，不适合原生运行资源密集型 MLLM。替代方法包括更小的、移动优化的

[1] 读作 "x-air"，我们利用云在 XR 体验中植入（少量）人工智能（X ai→← R)

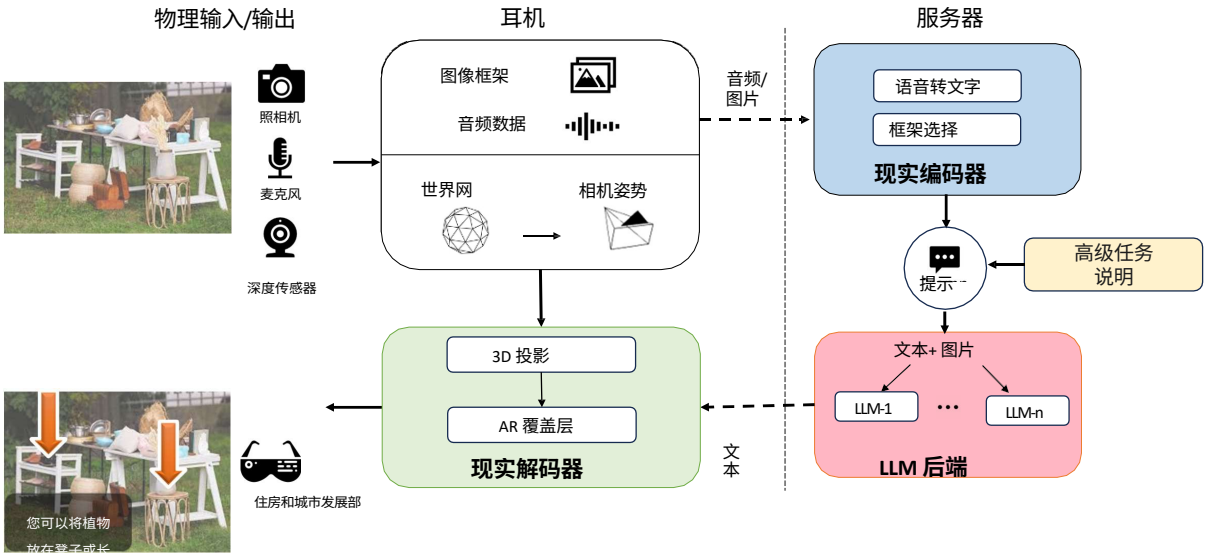


图 2：高层 XaiR 系统架构和数据流。图中显示了系统的不同组件、运行的设备以及捕获和通过网络传输的不同类型数据。

LLMs[35]，但它们往往为了性能而牺牲了准确性。因此，许多基于 LLM 的应用依赖于云架构来卸载密集的计算[1, 11]。

XaiR 采用了类似的分离式处理架构，以解决 MLLM 空间理解能力有限和 XR 头显处理能力受限的问题。它将 MLLM 推理等复杂操作卸载到附近的服务器上，而头显则在本地处理三维世界网格操作。该系统能有效管理多个模型的并发执行，这些模型处理以自我为中心的图像、语音和文本输入，所有这些都与世界网格中带有时间戳的姿态数据同步。处理任务的这种分离使系统能够将 AR 内容适当地投射到场景中，使输出文本与用户的时间、位置和方向保持一致。利用这些数据和 LLM 文本重新响应，虚拟内容通过光线投射从记录的相机姿势整合到耳机生成的 3D 地图中。我们的系统由三个主要部分组成：*现实编码器*（*Reality Encoder*），用于收集物理环境数据（姿态时间戳图像和音频）；*LLM 后端*（*LLM-backend*），用于合成各种专用 MLLM 的输出；以及*现实解码器*（*Reality De-coder*），用于完善文本输出并叠加 AR 注释。*高级任务描述*用作提示，指导 LLM 后端为特定任务定制系统（见图 2）。

我们使用一个 "认知助手 "应用程序对我们的平台进行了评估，该应用程序旨在利用 AR 注释指导用户完成详细的物理任务。该应用可处理涉及连续步骤的任务，例如指导用户组装工业机械或指导用户使用新的收费机。认知助手应用主要分为两个阶段。首先，专家用户引导系统完成任务，并要求 MLLM 从自我视频流中生成指令。其次，在实时会话中，系统使用预先生成的指令作为 "高级任务描述"，并使用捕捉到的自我中心图像作为参考，指导新手用户完成相同的任务。

我们的评估包括一项研究，旨在比较 MLLM 与人类向导在远程计算机上对同一数据源进行操作的性能，从而建立人类基线。参与者执行了各种物理任务，其中一些由我们的 MLLM 系统引导，另一些则由人工后端引导。我们使用任务完成时间、用户查询次数等指标，以及用于衡量认知负荷的 NASA 任务负荷指数（TLX）[12]，对每个向导的有效性进行了评估。此外，我们还分析了

分析输入数据的类型和数量如何影响 MLLM 提供分步指令的效率和准确性。总的来说，人类的表现优于 MLLMs，但有时两者表现非常接近。

总之，本文的贡献如下：

- 用于开发 XR 头显应用程序的开源系统，可将空间数据与 MLLMs 集成[2]。
- 根据人类基准，对 MLLM 在支持物理任务方面的有效性进行比较研究。
- 深入了解 MLLMs 有效支持现实世界任务需的最佳数据类型和数量。

2 相关工作

我们将相关工作分为两类：大型语言模型及其与 XR 的整合，以及基于指令的认知助手的前期工作。

2.1 XR 大语言机型

随着 LLM 的发展，我们看到越来越多的模型既具有通用性，又具有类似人类的理解和反应能力[10, 21]。由于 LLM 现在具有多模态能力，它们还可以对图像和音频进行零点推理 [28, 30]。这自然而然地导致了与 AR 应用的整合，它们的先进功能可以增强用户在沉浸式环境中的体验和互动。

然而，这些功能强大的 LLM 是资源密集型的，由于其对计算和内存的高要求，无法在移动 AR 设备上高效运行。随着通用 LLMs 开发的不断进步，人们已将大量精力用于使其能够在移动平台上运行。如 Zhang 等人的 TinyLLAMA [35] 和 Hsieh 等人的 Distilling Step-by-Step [13]，这些模型正在进行量化和稀疏化，从而能够在移动 XR 头显等设备上部署。虽然这提高了内存使用率、降低了延迟并减少了网络需求，但其代价是降低了描述性、准确性和一般应用性[14]。，GPT-4 [21]、Llama [5] 和 LLaVA [18]等强大的 LLM 拥有数十亿或数万亿个参数，并在庞大的数据集上接受过训练，因此能够在广泛的顶层设计中提供更详细、更富有想象力的响应。然而，它们的规模使其无法在边缘设备上运行，需要在功能强大的服务器上运行。识别

[2] Github 链接：https://github.com/srutisrinidhi/XaiR

针对这些限制，我们的系统在云中执行 LLM 处理，同时在耳机上处理 XR 相关任务，以平衡性能和资源使用。我们承认这种方法会增加延迟，但我们相信随着技术的进步，延迟最终会降低。延迟和响应质量之间的权衡是未来需要探索的一个领域，有可能会一种能够适应不断变化的网络或应用需求的系统，类似于 [19] 中的方法。

下面我们将讨论将 LLM 与物理数据连接起来的早期工作。OpenAI 的 GPT-4 omni [6]是与具有多模态的 LLM 进行更自然交互的一个步骤，它能够理解用户的上下文。Xu 的穿透式人工智能[31]展示了 LLM 如何将常识性知识应用于传感器数据，从而推断出更高层次的概念。我们的系统应用了类似的技术，但将它们与 XR 输出的以自我为中心的数据更紧密地联系在一起。

2.2 教学跟踪和认知助手

我们通过一个指令指导 "认知助手 "应用程序展示了 XaiR 的潜力，该应用程序可与许多现有方法进行比较。尽管在开发此类 XR 指令遵循助手方面取得了长足进步，但其中许多系统都需要大量人工来为每项特定任务制作 XR 指令[24, 33]。这些过程都是劳动密集型的，而且缺乏对不同任务的适应性。此外，这些系统通常提供固定的响应，缺乏根据用户交互动态调整指导的能力。虽然 Chi-dambaram 等人 [9] 的工作有助于指令生成，Stanescu 等人 [23] 的工作通过对象检测技术检测任务完成情况，但这两项工作在根据用户交互调整引导方面仍然存在不足。Step Check [4] 等商业系统将人工智能与 XR 结合起来，用于生产检查。虽然这种方法能自动检测错误，但仍需要花费大量精力为特定的制造流程创建每个人工智能模型。

LLM 在为编程、个人任务和医疗诊断等不同领域的虚拟助手提供支持方面展现出巨大潜力[2, 15, 20, 27]。然而，最近在多模态人工智能和感知技术方面取得的进展，开创了 "智能 "助手的新时代，这些助手能够解读物理环境并作出有针对性的响应。例如，Meta 公司的雷朋智能眼镜[1]利用人工智能推理来分析眼镜捕捉到的图像，并能深入了解用户周围的环境。同样，谷歌的阿斯特拉项目（Project Astra）[7]也能通过视觉和音频提示了解用户所处的物理环境并做出反应。虽然这类系统在感知环境方面表现出色，但它们仅限于显示固定的静态输出，缺乏动态投射和将响应锚定到环境中的能力。

在围绕开发由 LLM 驱动的物理任务助手的讨论中[8]，我们的工作代表了利用当今技术实现此类平台的初步努力。我们的目标是通过创建一个将 LLM 功能与现实世界交互相结合的平台来缩小差距，为理解物理环境的智能辅助工具奠定基础。虽然这不是我们工作的重点，但只需几个高级指令提示就能快速创建指令原型，这与以前的方法大相径庭。

3 系统设计

图 2 显示了三个主要软件组件（Reality Encoder、LLM-backend 和 Reality Decoder），它们分别客户端耳机和附近的服务器之间。客户端是在 Magic Leap 2 [3] 上运行的用 Unity [26] 开发的安卓应用程序、

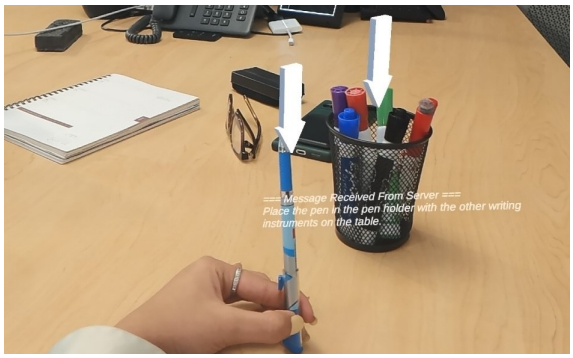


图 3：XaiR 为回答用户问题而生成的叠加在物理世界上的虚拟 AR 注释示例。取自 ML2 的屏幕截图。通过耳机，打开分段调光后的 3D 内容更加生动。

它可以串流音频、彩色图像和深度数据。客户端可以构建环境的三维世界模型，用于将图像中的物体投射到三维位置。服务器拥有更多的资源（我们使用的是一台 Linux 工作站，配备通过 NVLink 连接的双 RTX 3090 Ti GPU），可以执行音频到文本的翻译、对多达 130 亿个参数的模型进行 MLLM 推断以及文本后处理等任务。这种分工确保了所有 3D 处理都在耳机上进行，而与 MLLM 相关的文本和图像处理则委托给功能强大的服务器。

3.1 现实编码器

Reality Encoder 负责将音频转换为文本，并捕捉应传递给 LLM 后端的相关帧。在我们目前的实现中，数据从耳机传输到服务器上运行的现实编码器（以帮助调试），但该组件也可以在客户端运行，以减少网络开销。我们使用 OpenAI 的 Whisper 模型 [22] 将音频转录为文本，作为用户输入传递给 MLLM。音频数据被持续接收、解码，并在服务器上排队，以便进一步后处理。识别和过滤转换成分本身就是一个极其困难问题。作为一个简单的启发式解决方案，我们采用环境噪声调整，并在监听查询时使用 3 秒钟的窗口（即假设查询不会持续超过 3 秒钟）。在认知助手应用中，我们对转录的文本查询进行了扩充，使其包括当前正在执行的指令集等补充信息。

来自 ML2 的图像帧都标有唯一的标识符、时间戳和相机姿态，表明它们在传输到服务器之前的拍摄位置。摄像机姿态由 ML2 的跟踪算法确定，计算其相对于跟踪空间本地原点的位置。然后将这些帧添加到用户文本查询，以获取上下文信息，并转发到 LLM 后台。ML2 使用 Unity 中的 Magic Leap `XRMeshSubsystem` 定期生成 3D 地图，并将其存储在本地，以便日后进行 3D 内容锚定。由于用户的头部姿势和环境不断变化，三维地图以大约 1Hz 的速度定期更新。这一决定既节省了带宽，又保持了三维处理的本地化，并受益于 ML2 的内置功能。带有时间戳的姿势可实现可变时间处理，允许在处理完成后将场景中标注的位置锚定回三维世界地图中。

ML2 通过 WebRTC [29] 传输音频流，并使用 HTTP POST 请求发送上限为 640x480 分辨率的图像。为保持最佳推理质量，图像使用 PNG 进行有损压缩。默认情况下，图像的发送间隔为 1Hz、

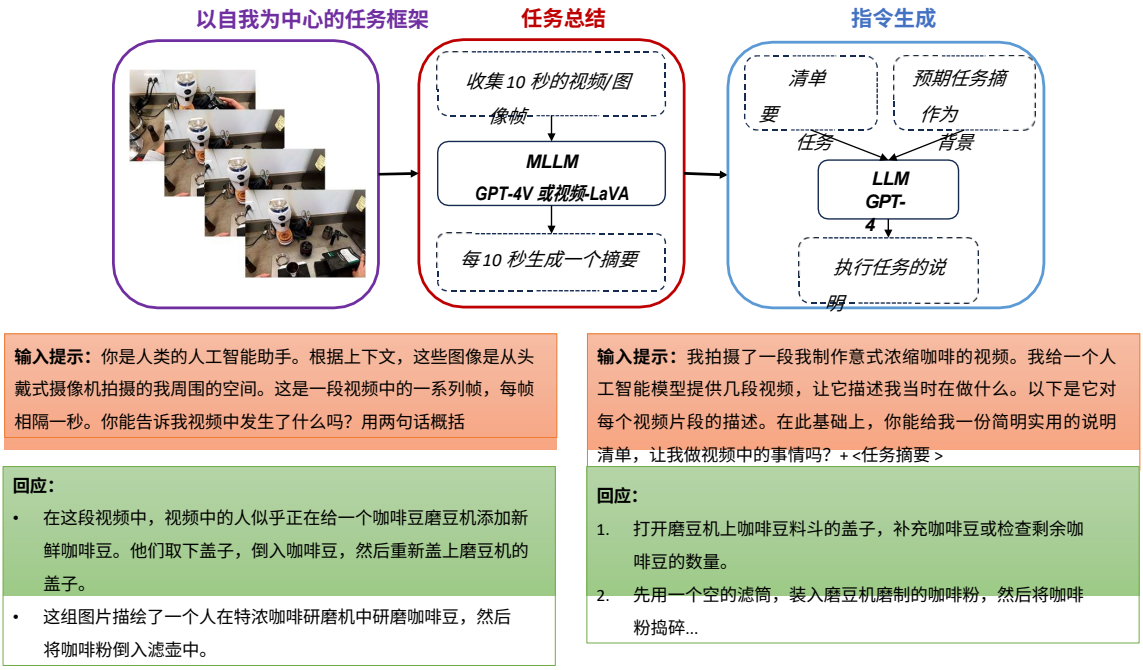


图 4：使用 XaiR 系统的认知助手指令生成流水线的详细分解。图中显示了输入提示示例和 MLLM 在每个步骤的输出响应示例。

在不断传输音频的同时。我们将在第 5.3.3 节中介绍 XaiR 的带宽要求。通过这种方式发送数据，后端可以轻松实现模块化和互换。

在我们的认知助手演示应用中，我们在每次提示时都会提供两个图像帧，这样我们就能为 MLLM 提供当前发生的事情的上下文，以及用户周围之前发生的事情的快照。虽然我们尝试过在 MLLM 加入两帧以上的图像，但我们发现这会大大降低系统的运行速度，而且不会显著提高响应的准确性。我们发现，这些值往往需要针对特定应用进行调整，并在第 5.2 节中对准确性和延迟进行权衡。

3.2 基于 LLM 的后台

在现实编码器步骤之后，生成的提示将被发送到我们的 LLM 后端。与耳机相比，服务器拥有充足的计算资源，因此我们提供了一个应用程序接口（API），可以并行执行多个 MLLM 以及潜在的远程云调用。在我们的基线系统中，我们同时使用 OpenAI 服务器上的 GPT- 4V [36] 和 Ferret [32]。Ferret 是一种专门的 MLLM，它可以识别物体，辨别图像中多个区域之间的关系，并提供物体位置的二维边界框。尽管 Ferret 的空间理解能力更强，但由于 Ferret 缺乏 GPT-4V 的高级推理能力，因此我们还是依靠 GPT-4V 进行推理。我们同时对两个模型进行查询，服务器结合两个模型的响应，使用 GPT-4V 的响应来提供文本反馈，使用 Ferret 的响应来提供用于锚定 AR 内容的物体位置。因此，我们的推理时间受限于最慢的 MLLM，通常是 GPT-4V。使用 GPT-4o 等速度更快的模型有可能减少这种开销。

3.3 现实解码器

现实解码器负责将 LLM 后端的输出打包成直观的视觉界面。我们使用一种简单的 AR 注释语言，该语言由我们的提示系统提供，可以绘制锚定在 3D 位置的图形基元。我们的原型系统有一个包含箭头和文本框的小型字典，但它可以很容易地扩展到包括更复杂的模型和最终的小型脚本交互。

主要挑战之一是将二维坐标转换为三维坐标。例如，Ferret 生成的 2D 方框需要投射到 3D 中，以便将 AR 内容锚定到场景中。ML2 会将所有先前的相机姿势及其相关图像 ID 保存在一个查找表中。一旦 ML2 从服务器收到带有图像 ID 的响应，它就会查找相关的相机姿势。利用该姿态，ML2 会向存储 3D 网格发出射线，以获取拍摄图像时该物体的 3D 坐标。文本输出也会以 AR 的形式显示在屏幕上，供用户使用。可视化效果见图 3。由于头显可在本地跟踪空间内对每个虚拟三维物体进行内部跟踪，因此我们无需持续处理每张图像来更新物体位置。一个限制因素是，由于在捕获帧后不会主动跟踪物体，因此锚点渲染的位置是固定的，只有在处理新帧后才会更新。如果物体在移动，这可能会造成明显的滞后。我们可以想象在客户端添加一个后处理步骤，对某些已知的动态物体进行持续跟踪。

4 XaiR 认知助理演示器

我们将我们的系统评估为认知助手[8]，具体来说，它可以引导非专业用户逐步完成指令，并能利用 XR 内容突出显示物理世界中的重要物体。我们的认知助手应用分为两个主要阶段--新任务的指令生成和指导用户完成任务的实时反馈模式。

4.1 从以自我为中心的视频中生成指令

图 4 展示了我们的系统如何从单个自我中心视频中自动生成指令。专家用户可以戴上耳机，录下自己执行任务的过程，然后使用 XaiR 提取分步文本指令。我们利用 MLLMs 解释文本和视觉数据的能力，从自我中心视频流和语音中生成专家的行动日志。收集到的图像可提供额外的视觉背景，说明特定对象和工具是如何用于某项任务的，而这些数据并不总是现成的 MLLM 所能知晓的。我们从 10 秒一批的图像流中提取帧，并使用 GPT-4V 将这些部分中执行的操作总结为文本。然后，我们收集所有文本并将其输入 GPT-4V，生成整体指令

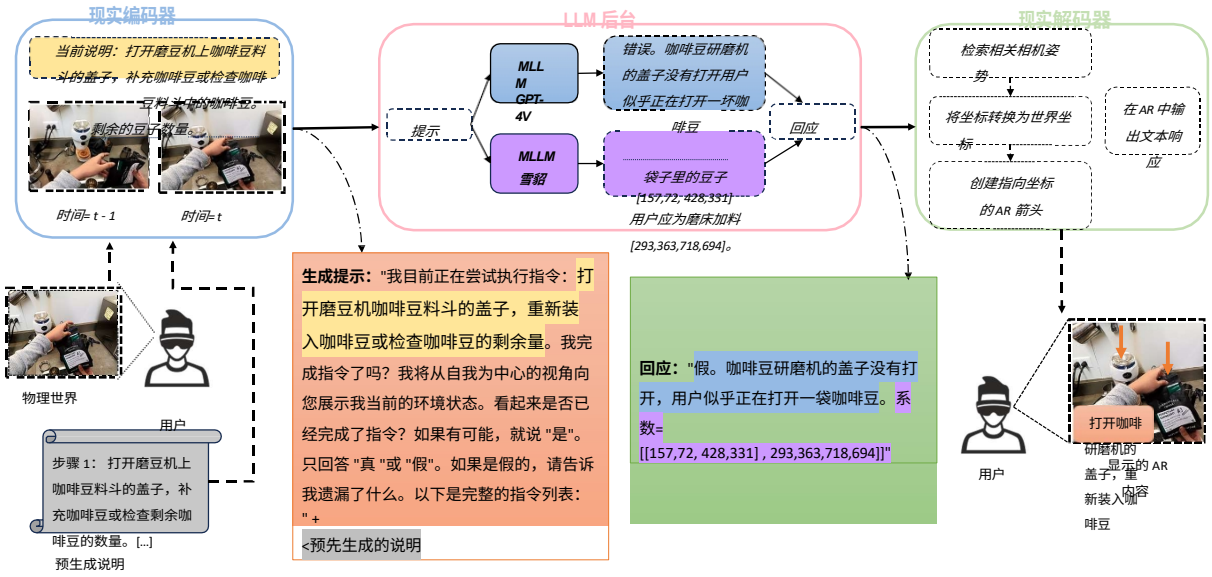
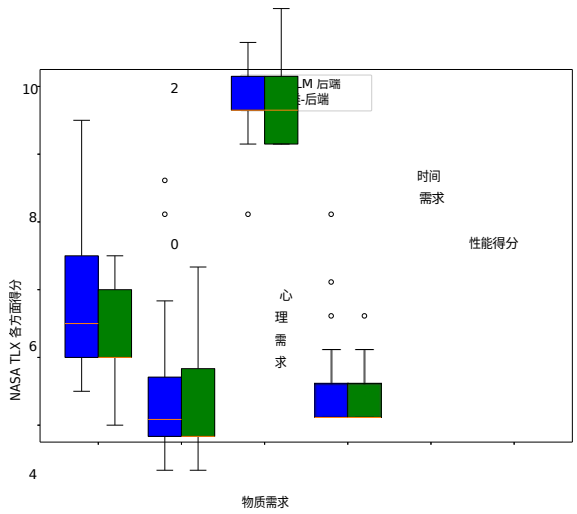


图 5：认知助手的指令跟踪流水线演练。该流程使用图 4 流程细节中生成指令。图中显示了生成的中间提示和 MLLM 响应

为非专业用户执行任务。这就实现了结构生成和用户操作记录过程的自动化。图 4 显示了我们的实际输入提示和系统生成的各种响应。

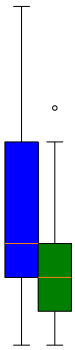
虽然 MLLM 能够纯粹根据文本输入创建完成任务的分步指令，但使用这种视频汇总和动作记录方法，可以确保生成的指令与专家完成任务的方式和使用的工具相匹配。例如，虽然 GPT-4V 可以轻松生成制作一杯咖啡的指令，但其结果往往非常笼统，与用户可能使用的特定工具无关。使用我们的技术，该模型将生成能够正确使用工具的说明。这在使用专用设备（如工厂）或使用特定方法完成任务时尤其有用。它还简化了指令的生成，因为专家不必手动写下完成任务的步骤，也不必冒着遗漏细节的风险，而 MLLM 可以捕捉到这些细节；它还消除了对 LLM 在特定任务中进行再训练的要求，使我们的系统具有很强的通用性。

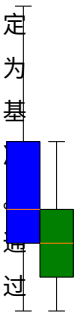


4.2 执行任务时的实时反馈

当非专业用户想要执行相同任务时，XaiR 可以使用前一阶段生成的指令引导用户完成任务。为了给 MLLM 创建提示，我们使用了最新的两个图像帧以及当前指令。提示要求 MLLM 利用图像帧作为用户所做操作的上下文，提供当前指令的状态。我们还在每个提示中提供整个指令列表作为上下文，这样 MLLM 就能获得关于已完成和即将完成任务的信息，从而提高对用户状态的理解。

ML2 会在环境中叠加虚拟箭头，以突出显示任务所需的重要对象。我们的系统还能生成文本，显示完成任务所需的反馈以及用户可能做错的地方。图 5，图中是这方面的一个详细示例的提示示例是由现实编码器和预先生成的指令组合而成的。在整个过程中，用户可以提出后续问题，这些问题也会被注入 GPT-4V 和 Ferret 的查询中。这样，用户就可以提出澄清问题或获得助理提供的补充信息。



MLLMs 能够平等地获取信息，我们开发了一个基于网络的仪表盘，7 所示。该仪表盘显示五个理世界数据、根据图像解释用户操作以及提供任务指导的能力。在研究过程中，为了确保人类助手和的性能。我们将人类视为 "理想 "模型，将其设定为基  通过比较，我们评估了 MLLMs 理解物

5 评估力

在接下来的章节中，我们将评估 XaiR 作为指导认知助手的有效性。

5.1 与人类相比的推理能

我们进行了一项用户研究，以评估 GPT-4V 和 Ferret 在认知助理任务中与人类的表现相比

图 6：NASA TLX 值按各个维度分列。图中比较了用户在对 MLLM 后端和人类后端认知助手进行用户研究时给出的分数。

每秒从用户的耳机摄像头捕捉图像。人类助手键入回复并在最新图像上绘制边框，这与 Ferret 子系统的功能类似。人类助手的文字回复包括对当前任务的反馈，以及是否进入下一个任务段的决定、

	基本任务		高级任务	
	MLLM	人类	MLLM	人类
准确度（%）	86.7	100	93.3	100
每条指令的查询次数	8.05± 2.98	2.52± 0.43	10.72± 3.80	3.88± 0.49
每条指令的时间（秒）	38.37± 3.16	27.88± 1.51	37.68± 3.10	27.85± 2.34
完成任务的时间（秒）	166.26± 15.81	139.4± 7.55	228.38± 13.63	162.93± 12.73
NASA TLX 原始分数（满分 100 分）	28.11± 3.08	16.67± 2.06	33.00± 4.41	17.78± 2.62

表 1：用户研究结果：比较 MLLM 后端与人类后端认知助手在 "基本 "和 "高级 "任务中的表现。数值显示的是 15 位参与者的平均成绩。对于 TLX 分数，平均值越低越好。

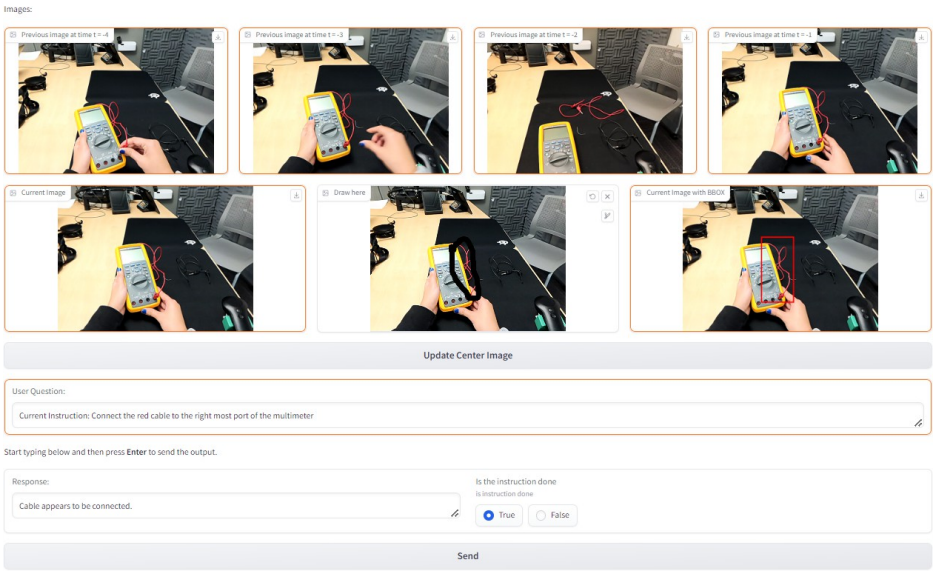


图 7：人类后端仪表板界面。它显示了一个序列中的前五个帧，并包含一个草图板用于标注 AR 叠加的位置。它还包括用户输入提示、用于回复的文本框以及用于决定当前步骤是否已完成的复选框。

与 GPT-4V 子系统的功能类似。为了进行公平比较，向人类和 MLLM 提供了相同的数据--相同的图像集和文本提示。人类推理代理是从作者中挑选出来的，在所有试验中保持不变，以确保一致的键入和注释速度，从而消除特定人类推理代理在各项研究中的瓶颈。

我们的研究涉及 15 名 22 至 45 岁的参与者，男女比例为 46%。每位参与者总共要完成四项任务：两项 "基本 "任务和两项 "高级 "任务。基本任务要求参与者根据提供的说明将简单的儿童积木组装成不同的结构。高级任务包括设置加湿器和使用万用表测量通孔电阻器的电阻。我们从每类任务中随机分配一项任务，由 GPT-4V 和雪貂 MLLM 指导完成，而另一项任务则由人工通过网络仪表板指导完成。直到研究完成后，参与者才会被告知有不同的指导代理。对于每项任务，我们都使用 NASA 任务负荷指数（TLX）记录了完成时间、查询次数和用户体验。结果见表 1。1.

我们发现，在使用 MLLM 执行的 30 项任务中，有 3 项任务是不完整的，这意味着 MLLM 在超过 4 分钟的时间内没有从一个步骤前进，即使该步骤已经完成，或者 MLLM 在错误地从一个步骤前进，即使该步骤

未完成。在这种情况下，我们将任务标记为 "未完成"。这样，基于 MLLM 的认知助手的总体准确率为 90%，而基于人类的认知助手的准确率为 100%。我们发现，与人类相比，参与者在与 MLLMs 交互时需要进行更多的询问，系统才会认为任务已经完成。我们看到，在使用 MLLM 后端执行基本任务时，每次指令平均需要查询 8.04 次，而人工后端只需要查询 2.52 次。与人类相比，MLLM 对每条指令的查询次数要多出 6-7 次，这表明 MLLM 的响应并不总是准确的（T 检验的 p 值= $1.56x10^{-5}$< 0.05）。我们观察到，如果系统没有检测到指令已经完成，参与者就会重新摆放面前的物体，或者尝试从不同的角度查看物体；参与者会自然而然地认为助手无法 "看到 "已完成的步骤，因此会移动物体来解释这一点。

虽然基于人类的认知助手在帮助用户完成任务方面速度更快，但我们看到，MLLM 在完成基本任务时多花了 19% 的时间，在完成高级任务时多花了约 40% 的时间，这表明当今 MLLM 的推理速度与人类相差不大。我们观察到，基于 MLLM 的试验将大部分时间多次调用 MLLM，而人类则需要更长的时间来查看图像、识别和定位相关对象以及键入文本回复。虽然我们起初认为人类花费的时间会明显少于 MLLM。

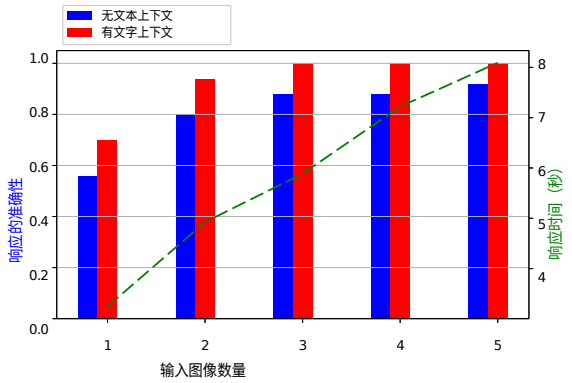


图 8：当提供不同数量的输入数据作为上下文时，GPT-4V 的准确性和速度。显示了响应精度和响应时间之间的权衡。

我们惊讶地发现，每次查询平均只相差 0.5-1 秒（T 检验的 p 值= 0.0065< 0.05）。为了评估用户体验，我们使用 NASA Task Load In- dex（TLX）来测量两种认知助手的用户工作量。表 11 表 显示，基于 MLLM 的助手的平均 TLX 得分始终高于基于人类助手，得分越低表明认知负荷越低。对图 6 中 NASA TLX 各组成部分的进一步分析表明，MLLM 支持的系统比人类支持的系统表现出更高的任务负荷指数。这表明，基于 MLLM 的认知助手需要用户付出更多的脑力劳动。尽管如此，性能指标几乎完全相同，这表明两个系统都有能力实现任务目标，只是用户需要承担不同的脑力劳动。

5.2 比较输入系统的数据值

现实编码器的目的是收集基本数据以便低 MLLMs 更好地了解用户的环境。为了确定我们需要向 GPT-4V 提供多少上下文信息才能实现最佳性能（在准确性和速度方面），我们尝试向 GPT-4V 提供不同程度的上下文信息，包括不同数量的图像和文本信息。在本实验中，提供的图像帧数从一帧到五帧不等，每个帧在任务期间的捕获时间间隔为一秒。此外，我们还提供了不同数量的文本上下文。例如，在一半的试验中，我们提供的文字上下文显示用户正在尝试煮咖啡，而在试验中，我们没有提供任何文字上下文。这些变化对任务预测准确性的影响如图 8 所示。我们观察到，随着提供给 MLLM 的图像帧数的增加，响应时间也在增加。为了在保持较低系统延迟的同时达到令人满意的准确性水平，我们得出结论，在 3.1 节中详细介绍的 "真实性编码器 "中使用两个图像帧和文本上下文是我们使用案例中最理想的上下文数量。

5.3 系统性能

5.3.1 AR 叠加的准确性

在 "现实解码器 "步骤（3.3 节）中，Ferret 会返回二维图像空间坐标，即我们应将虚拟叠加物光线投射到三维空间的位置

问题	回答问题需要：			AR 过度铺设精确
	从图像中识别物体?	从中识别物体?	寻找?	
问题 1：笔在哪里?	没有	没有	是	93.3%
问题 2：我可以用什么在白板上写字?	没有	是	是	83.3%
问题 3：我该把这支笔放回哪里?	是	是	是	76%

。为了衡量 Ferret 的准确性，我们向认知助手提出了一个口头问题，询问用户对空间中物体的感知，比如 "笔在哪里？"，然后对所得到的叠加效果进行视觉评估。在评估过程中，我们提出了三个不同难度的问题，每个问题重复 30 次，并记录下视觉评估的成功率。表 22 表 显示了每个问题的 AR 叠加结果的准确率。我们可以看到，随着问题的复杂程度

表 2：在涉及不同理解类型的问题（识别对象是什么、根据描述在场景中找到对象以及根据描述识别对象）中，AR 叠加的准确性。

换句话说，当问题从简单地在场景中找到一个已知物体转变为了解物体的用途并知道把它放在哪里时，回答的准确性就会降低。

5.3.2 指令生成的准确性

如第 4.1 节所述，我们的认知助手可以从专家用户执行特定任务的自我中心视频中总结并理解用户的操作。为了测试 MLLM 在生成正确、清晰指令方面的有效性和准确性，我们让用户执行五项日常任务，并从中生成指令。这些任务的复杂程度各不相同：安装加湿器、用意式咖啡机煮咖啡、制作三明治、摆放餐桌和焊接电线。

我们使用 GPT-as-a-judge[37]将生成的指令与作者生成的地面实况指令进行比较。我们的步骤如下：我们拍摄以自我为中心的视频，提取图像，并按照第 4.1 节所述使用 GPT-4V 生成指令。然后，我们将生成的指令和地面实况结构图同时交给 GPT4，并询问它生成的指令中有多少与地面实况相匹配，有多少指令是错误的，以及有额外的指令是未指定的但没有错误。由于指令生成的非确定性，我们要重复这一过程 10 次，然后求出返回值的平均值，得出正确指令、错误指令和额外指令的数量。

我们用 Video-LaVA [16] 代替 GPT-4V 作为结构生成代理进行了同样的实验，因为 Video-LaVA 可以直接解释视频而不是图像集合。我们使用与 GPT-4V 生成指令时相同的自我中心视频，并重复上述步骤。使用这些模型生成指令的结果如图 9 所示。我们注意到，GPT-4V 的准确率在 50%-60% 之间，而错误指令在大多数下都低于 20%，因此生成的指令整体上是相当真实的。我们注意到，在生成正确指令方面，LLaVA 的表现不如 GPT-4V，这主要是因为它在更少的数据上进行了训练。据传，GPT-4V 拥有约 1 万亿个参数，而 LLaVA 只有 30 亿个，因此 GPT-4V 显然是更高准确度和响应质量的不二之选。因此，尽管 Video-LLaVA 是开源的，而且速度更快，我们还是决定使用 GPT-4V 作为 MLLM，为我们的认知助手系统进行图像帧推理。

5.3.3 系统基准

XaiR 每秒通过网络从客户端向服务器传输音频和图像帧流。服务器将文本流传回

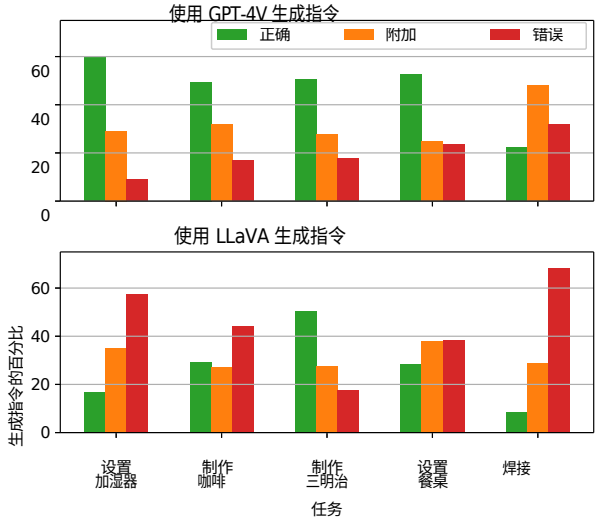


图 9：以自我为中心的视频生成的指令分析用户使用 **GPT-4V** 和 **LLaVA** 执行的不同任务。

客户端。与图像相比，文本和音频的数据量几乎可以忽略不计，因此我们主要根据图像帧流的需求来计算所需的大致带宽。每帧图像都是 640x480 的 PNG 压缩图像。研究期间消耗的平均带宽约为 **2.32 Mbps**。

系统组件	***所用时间（秒）***
端到端系统	4.241
MLLM 后端	4.173
GPT-4V	4.169
雪貂	2.883
现实编码器和解码器	0.067
图像流	0.005
RayCasting	0.001

表 3：各主要系统组件的平均延迟时间。

为了测量系统的端到端延迟，我们测量了从耳机捕捉图像到生成 AR 物体的平均时间。平均大约需要 **4.241 秒**。请参见表 3。3有关时间的详细信息，请参见表 。

6 讨论与局限性

XaiR 平台的一个局限是无法创建精确而复杂的增强现实（AR）叠加。我们使用 Ferret 为对象生成二维边界框。然而，Ferret 有 130 亿个参数，而且只在 110 万个空间接地数据点上进行过训练，因此缺乏像 GPT-4V 这样大型模型的鲁棒性，据说 GPT-4V 有大约 1 万亿个参数。因此，Ferret 检测复杂物体的准确性和能力不太可靠。此外，Ferret 还仅限于检测场景中明显存在的物体；它无法执行更复杂的空间检测，而这正是执行任务指令所必需的。例如，虽然它可以根据 "将笔移到书的右边 "的指令识别并提供笔和书的坐标，但它无法确定 "书的右边 "的坐标。因此，我们的 AR 叠加功能目前仅限于简单地突出显示相关对象，而无法演示如何通过更复杂的动画将我们的二维坐标投射到三维世界中。根据摄像头和深度传感器重建网格的过程非常缓慢（最快更新速率约为 1Hz）。因此，在程序运行的前 30-40 秒内生成的 AR 叠加效果往往会因网格重建不完整而显示出较低的深度值。此外，虽然网格生成对大型物体更有效，但深度摄像头的局限性意味着更小、更精细的物体往往会产生更多噪音和不正确的网格。因此，由于网格的问题和/或 Ferret 物体检测的错误，较小物体的 AR 叠加有时会不准确。

众所周知，大型语言模型（LLMs）由于参数数量庞大，其问题在于响应速度缓慢，推理通常需要数秒时间。这种延迟给系统带来了明显的延迟，在对话助手中尤为明显，用户可能需要等待 4-6 秒才能完成一个完整的对话。反应。这种滞后现象在涉及指令执行的任务中尤为严重，因为用户必须在收到指令后等待几秒钟才能做出反应。来执行任务，因为我们往往缺乏此类动画的精确空间位置。未来，随着 MLLMs 空间理解能力的提高，我们预计类似 XaiR 的系统在 AR 内容生成方面将取得重大进展。

此外，Magic Leap 2 头显虽然能够生成环境的 3D 网格，但也面临着一些限制，影响了

在 LLM 处理完成任务的图像之前，用户就已经完成了一项操作。因此，系统的速度本身就受到所采用的 LLM 速度的限制，而我们认为 LLM 的速度会随着时间的推移而降低。

最近，谷歌（Google）[7] 和 OpenAI [6]等公司都在努力将物理世界的数据整合到 MLLM 中，以帮助用户更好地了解周围的空间。他们的工作主要集中在改进 MLLM 如何仅使用音频/文本和彩色图像来回答有关用户周围环境的问题。我们用于场景理解的现实编码器/解码器架构将这一工作流程推广到了网格数据/世界信息（以及未来可能其他传感器模式）。我们提供了一个带有脚手架的开源架构，允许开发人员在 XR 应用背景下尝试使用 MLLM。此外，随着研究人员和工程师对 MLLM 和机器学习模型进行改进，这些新模型可以很容易地并行集成到我们的系统中，以提高性能和响应质量。有了我们的现实编码器和解码器系统，我们无需修改 MLLM 即可提供理解物理世界的能力以及地面响应能力，从而可以轻松测试新模型响应物理世界数据的能力。我们还相信，我们的 MLLMs 与人类用户研究方法可以作为评估未来系统和创建离线基准的重要工具。

7 结论

总之，本文介绍了 XaiR，这是一个将 MLLM 与 XR 相集成的研究平台，有望增强机器对物理环境的理解。XaiR 采用计算分离的方式，资源密集型 MLLM 操作在服务器上处理，而需要世界模型的操作则直接在 XR 耳机上管理，从而独特地促进了物理环境中多个 MLLM 的并发使用。我们通过认知助手应用程序演示了该系统，并进行了一项用户研究，以评估 MLLM 与人类操作员在任务完成率和时间方面的差异。结果表明，虽然 MLLM 对物理世界的理解和响应水平可能无法与人类相提并论，但它们处理任务的速度往往快于人类操作员。我们相信，这个框架可以用来生成数据集，这些数据集必然会被用来训练和评估未来的模型进展。

致谢

本研究部分得到了美国国家科学基金会（National Science Foun- dation）CNS-1956095、CNS-2148367、美国国家科学基金会研究生研究奖学金（DGE-2140739）和博世研究公司（Bosch Research）的资助。本材料中表述的任何观点、发现、结论或建议均为作者个人观点，与国家科学基金会无关。

参考资料

[1] Meta ray-ban glasses multimodal ai. https://about.fb.com/news/2023/09/new-ray-ban-meta-smart-glasses/ 。Ac- cessed：2024 年 1 月 2 日。1, 2, 3

[2] Github copilot。https://github.com/features/copilot/， 2023。在线。访问：2024 年 5 月。3

[3] 魔法飞跃。https://www.magicleap.com/en-us/，2023 年。在线观看。
已访问：2024 年 5 月。3

[4] 2024 年 7 月3

[5] Llama 3. https://ai..meta.com/blog/meta-llama-3/, 2024
在线。已访问：2024 年 7 月。2

[6] Openai gpt-4o。。https://openai.com/index/hello-gpt-4o， 2024在线。访问：2024 年 7 月。3, 8

[7] Project astra google deepmind. https://deepmind.google/technologies/gemini/project-astra/, 2024. 在线。Accessed：2024 年 7 月。1, 3, 8

[8] L.Cheng.介绍 Microsoft Dynamics 365 指南中的 copilot，11 月 2023。3, 4

[9] S.Chidambaram、H. Huang、F. He、X. Qian、A. M. Villanueva、T. S. Redick、W. Stuerzlinger 和 K. Ramani。Processar：基于增强现实的原位程序化 2d/3d ar 指令创建工具。In *Proceedings of the 2021 ACM Designing Interactive Systems Confer- ence*, DIS '21, p. 234-249.计算机协会、
DOI: 10.1145/3461778.3462126 3

[10] J.Devlin, M.-W. Chang, K. Lee, and K. Toutanova.Chang, K. Lee, and K. Toutanova.伯特：用于语言理解的深度双向变换器的预训练，2019 年。2

[11] J.A. V. Fernandez、J. J. Lee、S. A. S. Vacca、A. Magana、B. Benes、 和 V. Popescu。免提虚拟现实，2024 年。2

[12] S.G. Hart 和 L. E. Staveland.nasa-tlx （任务负荷指数）的开发：经验和理论研究的结果。*人类心理 工作负荷*，1（3）：139-183，1988 年。2

[13] C.-Y. Hsieh, C.-L. Li, C.-K.C.-Y. 、C.-L. Li 、C.-K.Yeh 、H. Nakhost、Y. Fujii、A. Ratner、
R.Krishna, C.-Y. Lee, and T. Pfister.Lee, and T. Pfister.Distilling step-by-step! out-performing larger language models with less training data and smaller model sizes, 2023.2

[14] V.Jain, L. Mei, and M. Verhelst.分析当代神经网络的能量-延迟-面积-精度权衡。In *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (*AICAS*)*, pp.2021.9458553 2

[15] C.Li, C. Wong, S. Zhang, N. Usuyama, H. Liu, J. Yang, T. Naumann、
H.Poon, and J. Gao.Llava-med：在一天内训练大型生物医学语言和视觉助手。In A. Oh, T. Naumann、
A.Globerson, K. Saenko, M. Hardt, and S. Levine, eds., *Advances in Neural Information Processing Systems*, . 36, pp.
Curran Associates, Inc.，2023 年。3

[16] B.Lin, Y. Ye, B. Zhu, J. Cui, M. Ning, P. Jin, and L. Yuan.Video-llava：通过投影前对齐学习联合视觉表示， 2023.1, 7

[17] H.Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee.Llava-next：改进推理、OCR 和世界知识，2024 年 1 月。1

[18] H.Liu、C. Li、Q. Wu 和 Y. J. Lee。视觉指令调整，2023 年。2

[19] V.V. R. M. K. Muvva、K. Samal、J. M. Bradley 和 M. Wolf。小型无人机系统的闭环感知子系统。在 *AIAA SCITECH 2023 论坛*上，第 2673 页，2023 年。3

[20] D.Nam、A. Macvean、V. Hellendoorn、B. Vasilescu 和 B. Myers。使用 llm 帮助理解代码。In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineer- ing*, ICSE '24.美国计算机协会，纽约州纽约市、
DOI: 10.1145/3597503.3639187 3

[21] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, and et al. Gpt-4 technical report, 2024.2

[22] A.Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I.Sutskever.通过大规模弱超级 视觉进行鲁棒语音识别，2022 年。3

[23] A.Stanescu, P. Mohr, M. Kozinski, S. Mori, D. Schmalstieg, and D.Kalkofen.用于增强型真实环境的状态感知配置检测

分步教程。doi: 10. 1109/ISMAR59233.2023.00030 3

[24] K.田中、Y. 藤本、M. 勘原、H. 加藤、A. 本木、K. 仓木 K.Osamura, T. Yoshitake, and T. Fukuoka.使用增强现实技术设计装配任务支持系统的指南和工具。In 2020 *IEEE International Symposium on Mixed and Augmented Re- ality (*ISMAR*)*, pp.00077 3

[25] G. Team, R. Anil, and S. B. et al. Gemini: A family of highly capable multimodal models, 2024.1

[26] Unity Technologies.Unity, 2005.在线。已访问：2024 年 5 月。3

[27] M.D. Vu、H. Wang、Z. Li、J. Chen、S. Zhao、Z. Xing 和 C. Chen。Gptvoicetasker：Llm-powered virtual assistant for smartphone, 2024.3

[28] W.Wang, Z. Chen, X. Chen, J. Wu, X. Zhu, G. Zeng, P. Luo, T. Lu、J.Zhou, Y. Qiao, and J. Dai.Visionllm：大型语言模型也是用于以视觉为中心的任务的开放式解码器。In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, eds., Ad- *vances in Neural Information Processing Systems*, vol. 36, pp.Curran Associates, Inc., 2023。2

[29] WebRTC 工作组。Webrtc, 2011.在线。访问： 2023 年 4 月。3

[30] J.Wu, W. Gan, Z. Chen, S. Wan, and P. S. Yu.多模态大型语言模型：调查。doi: 10.1109/ BigData59044.2023.10386743 2

[31] H.Xu、L. Han、Q. Yang、M. Li 和 M. Srivastava。穿透性人工智能：让智能手机理解物理世界，2024 年。3

[32] H.You, H. Zhang, Z. Gan, X. Du, B. Zhang, Z. Wang, L. Cao, S.-F. Chang, and Y. Yang.Chang, and Y. Yang.雪貂：在 任何粒度的任何地方引用和接地任何东西，2023。1, 4

[33] J.Zauner, M. Haller, A. Brandl, and W. Hartman.为分层结构编写混合现实装配指导书。In *The Second IEEE and ACM International Symposium on Mixed and Aug-ented Reality, 2003.*237-246, 2003. DOI: 10.1109/ismar.2003.1240707 3

[34] H.H. Zhang, X. Li, and L. Bing.Video-llama: An instruction-tuned audio-visual language model for video understanding, 2023.1

[35] P.Zhang, G. Zeng, T. Wang, and W. Lu.Tinyllama: An open-source small language model, 2024.2

[36] X.Zhang, Y. Lu, W. Wang, A. Yan, J. Yan, L. Qin, H. Wang, X. 、W.W. Y. Wang, and L. R. Petzold.Gpt-4v(ision) 作为视觉语言任务的通用评估器 , 2023.1, 4

[37] L.Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang、Z.Lin、Z. Li、D. Li、E. P. Xing、H. Zhang、J. E. Gonzalez 和 I. Stoica。用 mt-bench 和聊天机器人竞技场评判 llm-as-a-judge》，2023 年。7