无训练多模态大型语言模型编排

Tianyu Xie
Xiamen University
Xiamen
alexisty233@gmail.com

Yuhang Wu
Xiamen University
Xiamen
derekwu2020@stu.xmu.edu.cn

Yongdong Luo
Xiamen University
Xiamen
yongdongluo@stu.xmu.edu.cn

Jiayi Ji
Xiamen University
Xiamen
jjyxmu@gmail.com

Xiawu Zheng*
Xiamen University
Xiamen
zhengxiawu@xmu.edu.cn

ABSTRACT

不同的多模态大型语言模型（MLLMs）无法直接整合到一个统一的多模态输入输出系统中。在之前的工作中，由于模态对齐、文本到语音效率和其他整合问题的挑战，训练被认为是不可避免的组成部分。在本文中，我们介绍多模态大型语言模型编排（MLLM Orchestration），这是一种无需额外训练即可创建交互式多模态 AI 系统的有效方法。MLLM 编排利用大型语言模型固有的推理能力，通过显式工作流程协调专用模型，实现自然的多模态交互，同时保持模块化，提高可解释性，并显著增强计算效率。我们的编排框架建立在三个关键创新之上：（1）一个中央控制器 LLM 分析用户输入，并通过精心设计的代理动态地将任务分配给适当的专用模型；（2）一个并行的文本到语音架构，能够实现真正的全双工交互，具有无缝的中断处理和自然的对话流；（3）一个跨模态记忆整合系统，通过智能的信息综合和检索，在不同模态中保持连贯的上下文，在特定场景中有选择地避免不必要的模态调用，以提高响应速度。广泛的评估表明，MLLM 编排在标准基准测试中无需额外训练即可实现全面的多模态能力，相较于传统的联合训练方法，性能提高高达 7.8 %，延迟减少 10.3 %，并通过明确的编排过程显著增强了可解释性。我们的工作将编排确立为多模态系统联合训练的实用替代方案，为下一代 AI 交互提供更高的效率、适应性和透明性。

最近在大型语言模型（LLMs）方面的进展已经使多模态能力变得越来越复杂。GPT-4o 的发布从根本上改变了多模态人工智能的格局，展示了同时处理多种模态的可行性和优势，这激发了开发全模态模型的显著兴趣。这种对统一多模态处理的追求自然地符合人类的直觉 - 就像人类在日常交流中无缝地整合视觉、听觉和文本信息一样，这些模型旨在在一个框架下统一处理视频、图像、文本和音频输入。这样的统一处理不仅比传统的单模态或双模态方法提供了更流畅和高效的交互，还通过利用模态之间的互补信息，实现了更全面的信息理解和生成。这一方向的光明前景吸引了研究界的广泛关注，各种全模态模型力求实现真正自然和直观的多模态交互。

全模式能力的发展经过了几个关键技术进步。在 GPT-4o 成功之后，研究工作主要集中在两个核心方面：模态扩展和自然交互增强。针对模态扩展，早期的尝试如 MiniGPT-4 [1] 通过两阶段对齐方法建立了基础技术，使用预训练的 BLIP-2 视觉编码器和一个轻量级投影层。LLaVA-NeXT [2] 通过引入统一的视觉表示学习框架进一步扩展了这一点，而 LLaMA-Omni [3] 和 RLAIF-V [4] 提出了用于处理多种模态的新颖架构。对于自然交互增强，如图 1 (a) 所示：VITA [5] 通过三阶段训练流程（双语指令微调、多模态对齐训练和多模态指令微调）率先实现了非唤醒交互和音频中断处理能力，而 HumanOmni [6] 通过开发专业的交互模式并收集超过 240 万个人类场景视频片段和 1400 万条指令数据点，专注于以人为中心的场景。此外，Baichuan-Omni [7] 采用包括多模态对齐和多任务微调在内的两阶段训练方法，为无缝跨模态对话流和上下文保持引入了创新方法，有效处理音频、图像、视频和文本内容。

然而，现有的全模式方法 [1, 8, 9, 2, 10, 11, 6, 3, 12] 主要依赖于重新训练和扩展单个基础模型以适应多种模式。这种范式存在两个关键限制。首先，它带来了大量的训练成本：对齐异质模式需要广泛的定制数据集和密集的微调，导致显著的人工努力和计算开销。其次，它表现出较差的可扩展性：修改基础模型或添加新模式通常需要完全重新训练，严重限制了对新场景和不断变化的用户需求的快速适应能力。

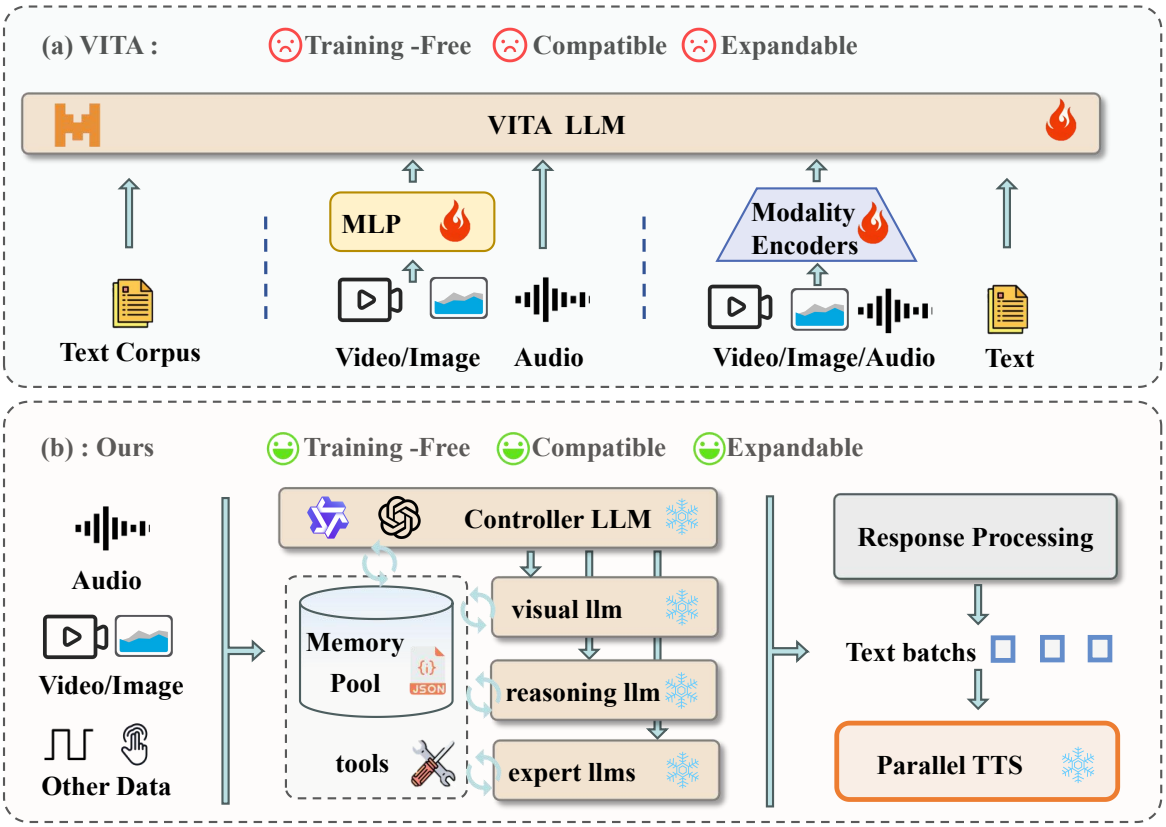


Figure 1: 展示了 VITA(a) 和我们的免训练多模态大型语言模型编排 (b) 的训练过程。我们的框架提供了一个免训练且高效的管道来处理全方位数据。

为了克服这些限制，我们采用一种新颖的视角，通过探索基于代理的方法来实现无需重新训练的全能能力。利用代理的内在灵活性和模块化，我们的想法是动态集成多个专门的模型，从而实现无缝的多模态交互。然而，简单迁移现有的基于代理的技术仍然面临几个独特的技术挑战：（1）分配：确定如何在不进行额外微调的情况下动态地将用户任务路由到最合适的专用模型。（2）记忆：促进在多个异构模型之间无缝共享对话上下文，实现连贯的多模态交互。（3）效率：确保响应迅速和实时交互，尽管协调多个专用模型引入了计算开销。

在本文中，我们开发了一种新颖的无需训练的多模态大语言模型编排（MLLM Orchestration）框架，如图 **??** 所示。具体来说，我们的框架由三个集成组件组成。首先，为了解决任务分配的挑战，我们引入了一个中央控制器 LLM，它利用其固有的推理能力，通过分层指令机制动态分析用户意图，并将任务分配给专门的专家模型。其次，为了解决记忆的挑战，我们提出了一个统一的跨模态记忆，它以标准化的 JSON 格式存储结构化交互历史，能够无缝检索和重用多模态上下文，以保持交互的一致性。第三，针对效率问题，我们开发了一个并行文本到语音（TTS）架构，通过基于语义的文本分割和并行批处理大大降低系统延迟，确保流畅和响应迅速的多模态对话。这些组件共同实现了无需训练的多模态交互，具有强大的可解释性、可扩展性和实时响应性。因此，我们的方法为用户提供了与一个由多个专门专家子系统动态组成的统一而智能的“超级模型”进行交互的体验。我们的方法避免了传统基于训练的范式固有的大量开销和僵化，为自适应和实用的多模态人工智能系统铺平了道路。

主要贡献可以总结如下：

- 无训练编排框架。一种新的多模态编排模式，通过智能调度和协调实现高效交互，消除了对大量训练或大型数据集的需求。
- 跨模态记忆整合。记忆整合机制将多模态信息统一为文本表示，支持在不同模态间无缝共享上下文而无需训练要求。
- 并行批处理 TTS 处理。这是一种高性能 TTS 架构，通过智能分块和缓冲实现接近实时的响应，显著降低感知延迟，同时保持输出质量。

- 实验结果。我们的框架在性能上与最新的基于训练的方法相比具备可比性或更优（例如，在 MMbench 上提高了 +1.93 %，在 Ai2d 上提高了 +2.73 %），同时通过并行处理优化将延迟减少了 10.3 %。

1 相关工作

1.1 全模态培训与多模态对齐

近年来，多模态大语言模型（MLLMs）在端到端训练技术的支持下取得了显著进展。在训练范式中出现了两种主要的技术方法：全参数训练和参数高效训练。其中，VITA [5] 作为第一个开源的互动全模态大语言模型，采用了创新的三阶段训练过程，包括双语指令微调、多模态对齐训练和多模态指令微调。基于 Mixtral 8x7B [13] 作为语言基础模型，它通过端到端训练实现了视频、图像、文本和音频模态的同步处理。

在参数高效训练方面，Freeze-Omni [14] 提出了一个新颖的冻结训练策略，该策略在保持语言模型参数不变的情况下，仅训练模态适配器。使用区区 60,000 条多轮对话数据，就实现了高质量的语音交互能力。这种方法不仅显著降低了计算资源需求，还有效避免了灾难性遗忘。诸如 Mini-Omni2 [15] 、LLaMA-Omni [3] 和 Moshi [16] 之类的工作也采用了类似的参数高效训练策略，为降低多模态训练成本提供了重要参考。

在开源领域，一系列具有代表性的多模态语言模型（MLLMs）已经出现，其中，PaLM-E 作为一种具身能力的多模态语言模型，在拥有 5620 亿参数的视觉-语言任务中取得了突破性进展；Flamingo 提出了一种新颖的少样本学习范式，显著提高了模型的泛化能力；BLIP-2 通过创新的两阶段预训练策略，在保持较少可训练参数的同时，达到了优秀的性能；LLaVA-NeXT 通过改进的推理能力和世界知识，在多个基准测试中超越了 Gemini Pro。VideoChat 和 VideoLLM 实现了高效的视频理解和生成；CogVLM 和 LAVIS 在视觉-语言理解方面取得了重要突破。HumanOmni 通过收集 240 万段视频片段和 1400 万条指令数据，致力于增强人类场景中的交互能力。此外，InternLM-XComposer-2.5、InternVL 和 Qwen2-VL 等模型在视觉-语言对齐和指令遵循上也取得了重要进展。Vary 和 Kosmos-2 进一步扩展了多模态模型的边界，表现出与专有模型相当的性能。

然而，与支持多种模态的专有模型如 GPT-4o [17] 和 Gemini [18] 相比，目前大多数开源模型仍然面临两个主要限制：首先，它们的模态支持范围有限，主要集中在图像-文本双模态 [19] ；其次，它们在用户交互体验方面的探索相对不足。这些限制严重制约了开源模型的实际应用。

1.2 基于代理的多模态系统

在代理系统领域，最近出现了一系列创新性的研究工作 [20, 21, 22, 23, 24] 。这些工作主要集中在两个方向：增强多模态能力和优化交互体验。AutoGen [20] 提出了一个多代理对话框架，通过灵活的代理交互模式和行为策略，实现复杂任务的自动分解和协同处理。该框架支持代理间的自然语言对话和代码生成，并能够有效整合 LLM、人类输入和外部工具。mmctagent [21] 设计了一种新颖的多模态代理架构，通过视觉-语言-行为的深度对齐与跨模态推理机制，增强代理在复杂视觉场景中的决策能力。LLaVA-Plus [22] 探索了代理的工具学习范式，提出了一种渐进式工具发现与使用机制，使代理能够根据任务需求自主选择并组合适当的工具。

在交互体验方面，LLaVA-Interactive [23] 通过统一的多模态交互框架实现了图像理解、生成和编辑功能的无缝集成。这项工作重点解决了模态转换过程中语义一致性问题，提出了一种基于对比学习的跨模态对齐方法。同时，最近的研究 [25, 26, 27] 揭示了多模态代理系统面临的三个主要挑战：大规模计算开销、模态语义对齐和实时交互延迟。为了解决这些挑战，研究人员提出了一系列优化解决方案，包括基于块的流处理、跨模态知识蒸馏和动态资源调度。这些工作为构建高效可靠的多模态代理系统提供了重要的技术支持。

在代理编排中，CrewAI [28] 专注于基于角色的代理编排，支持多代理之间的协作和任务分配。TaskWeaver [29] 提供了一个基于代理的任务自动化框架，使工作流管理更加灵活。在专业领域的应用中，ClinicalAgent [30] 提出了一个基于 LLM 的多代理临床决策支持系统，通过多代理协作提高了医疗诊断的准确性；LawLuo [31] 开发了针对中国法律咨询的多轮对话协作框架，通过四个专业代理（接待员、律师、秘书和监督员）模拟真实的法律咨询场景；InvAgent [32] 提出了一个创新的供应链库存管理多代理解决方案，通过 LLM 的零样本学习能力增强供应链网络的弹性和效率。在智能编排和优化中，Self-Organized Agents [33] 探索了超大规模代码生成和优化的 LLM 多代理框架，CMAT [34] 通过多代理协作成功提高了小语言模型的性能，而 Chain of Agents [35] 提供了一个针对长文本任务的高效 LLM 协作机制。

相比之下，我们提出的系统采用了与传统代理编排不同的方法，重点关注 LLM 的智能编排机制。通过创新的多模态 LLM 编排框架，该系统实现了视频、图像、文本和音频模态的集成，展现了高效率、灵活性和可扩展性。与现有的代理系统相比，我们的解决方案更轻量化且高效，不需要复杂的代理协作机制或训练，直接通过 LLM 编排实现多模态能力协调。通过系统的全面开源，我们希望为多模态 LLM 编排领域提供新的研究方向，并促进该技术在更广泛实际场景中的应用。在这一部分中，我们介绍了我们的多模态大型语言模型编排（MLLM Orchestration）框架背后的方法，该框架通过无需训练的方法实现了无缝的多模态交互。我们的方法由三个主要模块组成：MLLM 编排推理、跨模态记忆整合和并行文本到语音（TTS）生成。这些模块协同工作，实现了高效、动态的任务编排和多模态融合，增强了解释性、计算效率和灵活性。

问题定义。我们首先将多模态交互任务定义为一个过程，其中在第 t 回合给定用户查询 q_t 时，系统必须生成自然语言响应 o_t 并提供音频输出 a_t 。该交互涉及多种模态（例如，文本、视觉、语音）以及需要在各回合之间保持上下文。让 M_{t-1} 表示积累到第 $t-1$ 回合的记忆，该记忆储存相关的多模态信息。我们的任务是在确保模型能够选择适当的模态特定模型的同时，将响应 a_t 输出为语音合成结果。形式上，系统作为一个函数运行：

$$F : (q_t, M_{t-1}) \mapsto a_t \tag{1}$$

以前的方法通常通过广泛的端到端训练或参数高效的微调来整合多模态能力，这需要大量的计算资源和多模态数据。此外，这些方法通常在可解释性和灵活性方面存在挑战，因为任务需求的变化会导致昂贵的重新训练。如方程 (1) 所示，我们的系统将用户查询 q_t 和历史记忆 M_{t-1} 作为输入，并生成音频响应 a_t 。与以往基于训练的方法不同，我们提出了一种无需训练的编排方法，利用了控制器 LLM 的内在推理和规划能力。

我们的方法的完整流程在附录 A 的算法 1 中形式化，展示了不同组件如何交互以处理多模态输入并生成连贯的响应。我们的方法在三个关键步骤中处理输入：(1) 控制器 LLM 分析用户意图并选择适当的专用模型，(2) 跨模态记忆整合模块检索并整合来自 M_{t-1} 的相关历史信息，以及 (3) 并行 TTS 模块高效生成最终的音频响应 a_t 。这种模块化设计使得专用模型的动态编排成为可能，而无需端到端训练。

具体来说，我们的控制器动态地协调专业模型，包括视觉分析、文本推理和语音合成，而无需额外训练。这是通过将复杂的多模态交互分解为模块化工作流程来实现的，每个流程由不同的子模块管理：用于意图分析和模型选择的控制器 LLM（附录 A 中算法 1 的第 4-13 行）、用于保持上下文的跨模态记忆整合（第 14-21 行），以及用于高效响应生成的并行化 TTS（第 22-26 行）。

1.3 MLLM 编排推理

在接收到用户输入 q_t 后，我们的编排工作流程的第一步涉及控制器 LLM。这个控制器负责根据即时查询和累计的跨模态上下文 M_{t-1} 解释用户的当前意图，形成输入 $x_t = (q_t, M_{t-1})$ 。它输出一个令牌序列：$Y_t = f_{\text{ctrl}}(x_t) = [y_1, y_2, \ldots, y_L]$ ，其中每个令牌 y_i 可以是一个内容令牌（用于形成系统的自然语言响应）或者是一个驱动编排行为的特殊控制令牌。

为了实现这种无训练的编排能力，我们设计了一个结构化的提示指令框架，用来指导控制器的推理过程。完整的提示模板设计及其关键原则可以在附录 C 中找到。

需要注意的是，这只是一个模板示例。在实际应用中，我们的提示是在运行时由一个系统级的提示构建器动态生成的，该构建器会考虑当前对话上下文和可用的专家模块集合。具体而言，构建器维护一个注册表，记录专家能力——每种能力都与 `[S.need_*]` 形式的标签和相应的自然语言指令相关联。给定当前用户查询 q_t 和记忆上下文 M_{t-1} ，系统执行轻量级语义分类（例如，逻辑推理：[S.need_reasoning]，意图估计：[S.listen]）以确定可能涉及的模式。特殊控制标记遵循统一且可扩展的格式 `[S.need_*]` ，支持跨开放模式集和工具进行即插即用的编排。完整的指令集和详细使用指南可以在附录 B 的表 1 中找到。

控制器的输出序列 Y_t 被分为两部分：$Y_t = C_t \cup S_t$ ，其中 C_t 表示标准内容标记，$S_t \subseteq \mathcal{S}$ 则表示控制标记集。这些控制标记指导下一步的协调步骤。例如，当控制器识别出用户的问题涉及到一个视觉对象时，它可能输出：$S_t = [\texttt{[S.need\_vision]}]$ ，从而在下一阶段激活视觉模型模块。我们的 MLLM 协调框架完整算法详细见附录 A 中的算法 1。

1.4 跨模态记忆整合

一旦控制器 LLM 通过输出控制符号集 $S_t \subseteq \mathcal{S}$ 确定了所需的模态转换，编排过程就开始通过激活相应的专家模型来进行。这些模型需要访问相关的多模态输入，这些输入要么由用户直接提供，要么从之前的对话轮次中存储下来。为此，我们设计了一个跨模态内存池，用作存储和检索在整个对话过程中结构化的多模态上下文的统一知识库。

在每次轮到 t 时，当控制器输出诸如 `[S.need_vision]` 、`[S.need_speech]` 或其他特定模式的指令时，系统必须决定调用哪些专家模型。我们使用一个模式选择函数来形式化这个过程：

$$\delta_{\text{modality}}(x_t) = \begin{cases} \{m_1, m_2, \ldots, m_k\}, & \text{if } \exists\, \texttt{[S.need\_}m_i\texttt{]} \in f_{\text{ctrl}}(x_t), \\ \emptyset, & \text{otherwise}, \end{cases} \tag{2}$$

其中 $m_i \in \mathcal{M}$ 是一个注册的模态（例如，视觉、推理、音频），$\mathcal{M}$ 表示系统中可用的专家模块集合。此函数将控制器的输出映射到当前交互所需的具体模态要求。

对于每个选择的模态 $m_i \in \delta_{\text{modality}}(x_t)$，调用检索函数 h_{m_i}，从跨模态记忆池 M_{t-1} 中提取最相关的上下文或输入数据。例如，当需要视觉信息时，系统计算：$v_t = h_{\text{vision}}(q_t, M_{t-1})$，其中 v_t 表示与当前查询相关的检索到的视觉数据（例如，图像嵌入、场景描述或 OCR 结果）。为了通用性，我们将所有检索到的数据集表示为 $\{d_t^1, d_t^2, \ldots, d_t^k\}$，其中每个 d_t^i 对应于特定模态的数据单元。

检索到的数据通过一种模态感知的整合函数整合到 LLM 的推理流程中：$\widetilde{Y}_t = I(Y_t, \{d_t^1, d_t^2, \ldots, d_t^k\})$，其中 $\widetilde{Y}_t$ 是更新后的标记序列，它用实际的数据描述或摘要替换占位控制标记。因此，最终生成的答案变为：$O_t = C_t(\{d_t^1, d_t^2, \ldots, d_t^k\})$，这表示通过多模态输入提供信息的完整响应内容。

记忆池结构。跨模态记忆池在每轮对话时更新 t，以包括新的多模态观察、系统动作和对话条目。每个记忆项都以结构化的 JSON 格式存储（详细模式见附录 D），这使得在保留对未来模态的可扩展性时，能够高效存储和检索多模态信息。

记忆压缩。随着对话的推进，记忆池 M_t 会扩展，最终可能超过 LLM 的输入长度限制。为了解决这个问题，我们实现了一种内容感知压缩策略。我们定义了一个记忆压缩函数：$M_t' = h_{\text{compress}}(M_t)$，，该函数输出一个保留未来推理必要信息的简化记忆 M_t'。压缩受语义相关性、模态多样性、新近性和查询频率等因素的引导。例如，较早的图像描述及其相关的问答对可被摘要为抽象形式（如："用户询问图像中的警告标志。标志上写着：'前方危险'。"），而琐碎或冗余的条目则被丢弃。

这种策略确保了记忆在上下文上保持丰富但紧凑，在长时间的会话中保持连续性，同时维持计算效率。函数 h_{compress} 可以通过基于规则的启发式方法或具有摘要能力的小型辅助 LLM 进行实例化，这取决于部署的限制条件。

总之，跨模态记忆模块作为知识持久性、动态上下文融合和高效记忆管理的关键接口，使系统能够进行长距离多模态推理，而无需重新训练或数据丢失。

在控制器 LLM 完成调度任务并且相关的专家模型贡献了它们的特定模态输出之后，最终的文本响应 O_t 被组装完成。这个响应集成了多模态信息，代表了用自然语言给出的完整系统回复。工作流程的最后一步是将这个文本输出转化为语音，从而实现实时的人类般交互。这个过程由并行 TTS 生成模块处理。

为了优化延迟和流畅性，系统采用了基于分段的批量合成方法。在接收到最终内容标记 C_t 后，我们首先应用基于规则的分段策略，将文本分割成语义上连贯的块：$T = [T_1, T_2, \ldots, T_n]$,，其中每个段 T_i 都是一个独立的从句、句子或短语。详细的分段规则和示例可以在附录 E 中找到。

文本到语音（TTS）功能 $g(T_i)$ 被独立并行应用到每个片段 T_i：$a_{t,i} = g(T_i) \quad \text{for } i = 1, 2, \ldots, n$。为了降低延迟，我们采用流式合成和播放策略。当第一个片段 $a_{t,1}$ 合成后，它立即开始播放，同时后续片段以异步方式进行合成：$\text{play}(a_{t,i}) \parallel \{g(T_{i+1}), g(T_{i+2}), ..., g(T_n)\}$，其中 $\parallel$ 表示并行执行。最终的音频流 a_t 通过实时串接构建，并在片段边界进行节奏调整：$a_t = \text{stream\_concat}(a_{t,1}, a_{t,2}, \ldots, a_{t,n})$。由于用户在系统继续合成剩余片段的同时开始听到响应，这种流式方法显著减少了感知延迟。系统维护已合成片段的缓冲区，以确保平稳的播放过渡，同时 stream_concat 操作处理实时节奏调整，以在片段边界间维持自然的语音流畅度。

1.5 示例工作流程

考虑一种情景，其中用户展示了一张他们花园的图片并询问"这张图片中有哪些花正在开放？"在第 t_1 步，控制器 LLM 分析询问并输出 $f_{\text{ctrl}}(q_1, M_0) \rightarrow \{\texttt{[S.need\_vision]}\}$，触发视觉模型。系统检索视觉信息 $v_1 = h_{\text{vision}}(q_1, M_0)$ 并识别出各种花卉。生成的回应被分割为 $T_1 = \text{segment}(C_1)$，产生"我可以看到几朵玫瑰和郁金香盛开"，然后进入 TTS 流程。

当系统在说话时，用户打断道"有多少玫瑰……"控制器立即检测到中断模式，并执行 $f_{\text{ctrl}}(q_2, M_1) \rightarrow \{\texttt{[S.stop]}\}$，紧接着执行 $\text{clear}(Q_{\text{TTS}})$ 来停止当前的语音输出。由于问题不完整，它还会输出 `[S.listen]` 并更新内存 $M_2 = M_1 \cup \{(q_2, v_1)\}$。

用户完成了他们的问题"……这里有多少玫瑰？"系统使用缓存的视觉信息从内存 $v_1 \in M_2$ 中处理这个后续查询，而无需重新激活视觉模型。响应"图中有三朵红玫瑰"通过并行 TTS 合成，其中 $\text{play}(g(T_{3,1})) \parallel g(T_{3,2})$ 在准备后续片段的同时启用即时播放。

这种自然的交互流程展示了系统如何无缝集成视觉处理（h_{vision}）、内存管理（M_t）和流式语音合成，同时通过中断处理（$\text{clear}(Q_{\text{TTS}})$）保持响应的用户交互。

2 实验与结果

2.1 实验设置

我们的实验旨在验证我们方法的三个关键优势：（1）无训练编排的优越性相比于微调模型，（2）我们框架的灵活性和可扩展性，以及（3）我们流水线的透明性和可解释性。模型配置。我们使用 Qwen2.5-14B [36] 作为编排的控制器 LLM，并将各种模型整合为执行器，包括 Qwen2.5-VL 模型（7B，32B，72B）[37, 38] 、Qwen-VL-Max [39] 、Qwen2.5-Omni-7B [40] 、LLaVA-Video [41] 、InternLM-XComposer2.5-OmniLiveand [42] 、M2-omni [43] 、LLaVA-OV 模型 [44] 。我们还与商业解决方案进行比较，包括 Claude 3.5 Sonnet [45] 、Gemini-1.5-Pro [18] 、GPT-4V [46] 和 GPT-4o [17] 。由于现有的大多数多模态基准侧重于视觉任务，我们的专家模型配置主要利用 Qwen2.5-VL 模型进行专门处理。部分结果参考了上述官方论文。评估基准。我们在多样的多模态任务中进行评估：一般多模态理解（MME [47] ，MMBench [48] ）、视觉理解（MMStar [49] ，LVBench [50] ，Video-MME [51] ）、时间推理（MMMU [52] ）、数学推理（MathVision [53] ）、文本识别（CC-OCR [54] ）和图表理解（AI2D [55] ，InfoVQA [56] ）。

2.2 与主流全能模型比较

我们首先将无训练的 LLM 编排框架与最先进的 omni LLM 进行比较，以证明我们方法的有效性。表 1 与领先的多模态模型进行了全面比较。我们的方法在多个基准上始终优于所有竞争对手。在 MMBench-EN 上，我们达到了 88.54 % 的准确率，超过了 GPT-4o（83.10 %）5.44 %，超过了 Qwen2.5-Omni（81.80 %）6.74 %，以及 VITA（71.80 %）16.74 %。在 MMStar 的视觉理解上，我们的模型达到了 69.37 %，超过了 GPT-4o（64.70 %）4.67 %，超过了 Qwen2.5-Omni（64.00 %）5.37 %，以及 M2-omni（60.50 %）8.87 %。最显著的是，在时间推理（MMMU）上，我们的框架达到了 70.04 %，相比 GPT-4o 和 Qwen2.5-Omni（均为 59.20 %）提高了 10.84 %，相比 VITA（47.30 %）提高了 22.74 %。虽然 GPT-4o 在 Video-MME 性能（71.90 %）上领先，但我们的方法（65.58 %）仍然优于其他专用多模态系统。

Model	General Multimodal		Vision	Temporal	Efficiency
	MMBench-EN	Video-MME	MMStar	MMMU	Resp. Time (s)
GPT-4o	83.10	71.90	64.70	59.20	1.2
Qwen2.5-Omni	81.80	64.30	64.00	59.20	6.0
VITA	71.80	59.20	46.40	47.30	3.7
IXC2.5-OL	–	60.60	–	–	–
M2-omni	80.70	60.40	60.50	51.20	–
Ours	88.54	65.58	69.37	70.04	3.2

Table 1: 在一般理解、视觉、时间推理和效率指标方面与领先的多模态模型进行比较。

```
Finding 1: Training-free LLM orchestration strategy can effectively combine specialized
models, with capabilities approaching or exceeding commercial and open-source alternatives.
```

我们进一步将我们的方法与多种其他多模态模型进行比较，以验证其在不同场景下的有效性。

在通用多模态任务中，我们的编排方法在 MMBench-EN 上取得了 88.54 % 的准确率，达到了最先进的水平，超过了像 Qwen2.5-VL-72B 这样的更大模型。在视觉理解任务中，我们在 MMStar（69.37 %）和 LVBench（50.27 %）上展示了持续的优势，验证了我们编排策略在多样视觉理解场景中的有效性。值得注意的是，我们的资源使用分析显示，在实验中，32B 模型约占总推断调用的 60 %，其余则根据任务需求分布在其他专用模型之间。这表明与仅使用更大模型相比，我们的编排方法能够以显著减少的计算资源实现相当或更优的性能，突显了智能任务路由的效率优势。

```
Finding 2: Cross-modal memory pooling for context integration can enhance performance in
complex visual tasks that require context awareness.
```

Model	General Multimodal			Vision Understanding		
	MME	MMBench-EN	MMBench-CN	MMStar	LVBench	Video-MME
Qwen2.5-VL-7B	1673	84.45	84.98	59.94	45.30	56.62
Qwen2.5-VL-32B	1915	85.55	88.77	66.43	49.00	62.39
Qwen2.5-VL-72B	1980	86.61	90.44	68.22	47.30	65.74
Qwen-VL-Max	2281	77.60	76.40	–	–	51.30
Qwen2.5-Omni	2340	81.80	–	64.0	–	64.30
VITA	2006.5	71.80	–	46.40	–	59.20
LLaVA-OV-7B	–	80.80	–	61.70	–	58.20
LLaVA-OV-72B	–	85.90	–	66.10	26.90	66.20
InternVL-2-8B	–	81.70	–	59.40	–	–
InternVL-2-26B	–	83.40	–	60.40	–	–
Claude 3.5 Sonnet	–	–	–	–	–	–
Gemini-1.5-Pro	–	–	70.90	–	33.10	75.00
GPT-4V	517/1409	75.00	74.30	57.10	–	59.90
GPT-4o	2310.3	83.10	–	64.70	34.70	71.90
Ours	1922	88.54	89.35	69.37	50.27	65.58

Table 2: 在一般多模态和视觉理解任务上的性能比较，我们的方法在 MMBench-EN（88.54 %）、MMStar（69.37 %）和 LVBench（50.27 %）上表现优于其他方法。

2.3 视频理解性能

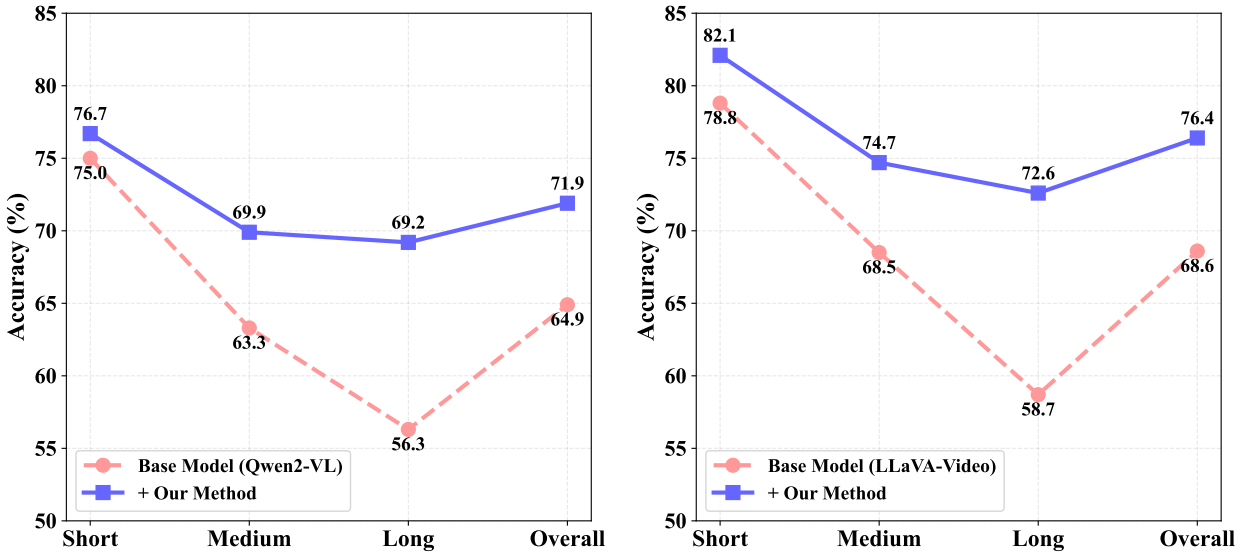
除了静态图像任务之外，我们还考察了我们的编排框架如何增强视频理解，这是一个具有复杂时间依赖性的领域。如图 2 所示，我们的框架在不进行模型微调的情况下实现了一致的改进。通过利用语音识别进行语音到文本的转换和多模态融合，我们观察到在所有视频长度上都有显著的提升。对于 Qwen2-VL，与基础模型（75.0 %，63.3 %，56.3 %）相比，我们的方法在短、中、长视频上分别取得了 +1.7 %，+6.6 % 和 +12.9 % 的提升，总体上提高了 +7.0 %（64.9 % 71.9 %）。类似地，对于 LLaVA-Video，与基础模型（78.8 %，68.5 %，58.7 %）相比，我们的方法显示了 +3.3 %，+6.2 % 和 +13.9 % 的改进，最终实现了 76.4 % 的最先进的整体性能（从 68.6 % 提高了 +7.8 %）。这些结果表明，我们的控制器在编排多模态处理和融合来自不同模态的信息方面的能力显著增强了视频理解能力，特别是在需要时间推理的复杂场景中。

```
Finding 3: Our framework integrates an additional text modality into inference through
a controller LLM, enhancing model inference performance and demonstrating that effective
multimodal fusion can be achieved without training.
```

2.4 专门任务表现

除了通用视觉任务之外，我们还在需要专业知识的特定领域评估我们的框架。我们的框架在需要专业知识的特定任务上表现出了强大的能力（表格 3）。在数学推理方面，我们的方法在 MathVision 上实现了 39.11 %，超越了 Qwen2.5-VL 模型和 GPT-4o（30.39 %）。在文本识别方面，我们在 CC-OCR 上达到了 79.11 %，与 Qwen2.5-VL-72B（79.80 %）相当，同时显著超过 Claude 3.5 Sonnet（60.63 %）和 GPT-4o（65.72 %）。

```
Finding 4: The controller LLM effectively leverages different specialized models based on
task requirements, allowing for targeted processing without requiring dedicated training for
each specialized domain.
```



(a) Performance Comparison on Qwen2-VL **(b) Performance Comparison on LLaVA-Video**

Figure 2: 在 Video-MME 基准测试上的性能比较。我们的编排机制通过有效利用自动语音识别技术进行语音到文本的转换以及多模态融合，在不同视频长度中实现了一致的改进。对于较长的视频，改进尤为显著，因为我们的控制器在整合音频信息时能够保持时间上的上下文，从而表现出最大的优势。

Model	MathVision	CC-OCR	AI2D	InfoVQA
Qwen2.5-VL-32B	38.40	77.10	72.26	83.40
Qwen2.5-VL-72B	38.10	79.80	76.25	87.30
LLaVA-OV-72B	–	–	85.60	74.90
Claude 3.5 Sonnet	–	60.63	94.70	49.70
GPT-4o	30.39	65.72	94.20	–
Ours	39.11	79.11	78.98	88.14

Table 3: 在需要专门技能的特定任务上的性能比较。我们的方法在数学推理（MathVision）、文本识别（CC-OCR）和图表理解（InfoVQA）方面取得了强劲的结果。

2.5 文本到语音处理效率

作为我们的第二个核心创新，我们分析了并行 TTS 处理架构的效率。如图 3 所示，我们基于批处理的并行处理方法有效地平衡了效率和语音质量。通过将长文本分割成适当大小的批次，我们将平均处理时间从 0.204 秒减少到 0.183 秒（减少了 10.3 %），同时提高了稳定性（方差从 0.056 秒降低到 0.013 秒）。

```
Finding 5: The parallel batch execution strategy enhanced stability maintains consistent
voice quality and natural speech rhythm, with the batch based method proving superior for
longer text sequences where natural prosody is essential.
```

2.6 兼容性声明

值得注意的是，这里报告的性能并不代表我们框架的上限，因为每个组件都能无缝替换为更强的模型。模块化设计允许在不进行架构更改的情况下，轻松集成新的模态和更强大的专用模型。这些结果验证了我们的无训练大型语言模型编排方法为构建更高效和适应性更强的多模态人工智能系统提供了一个有前景的方向。

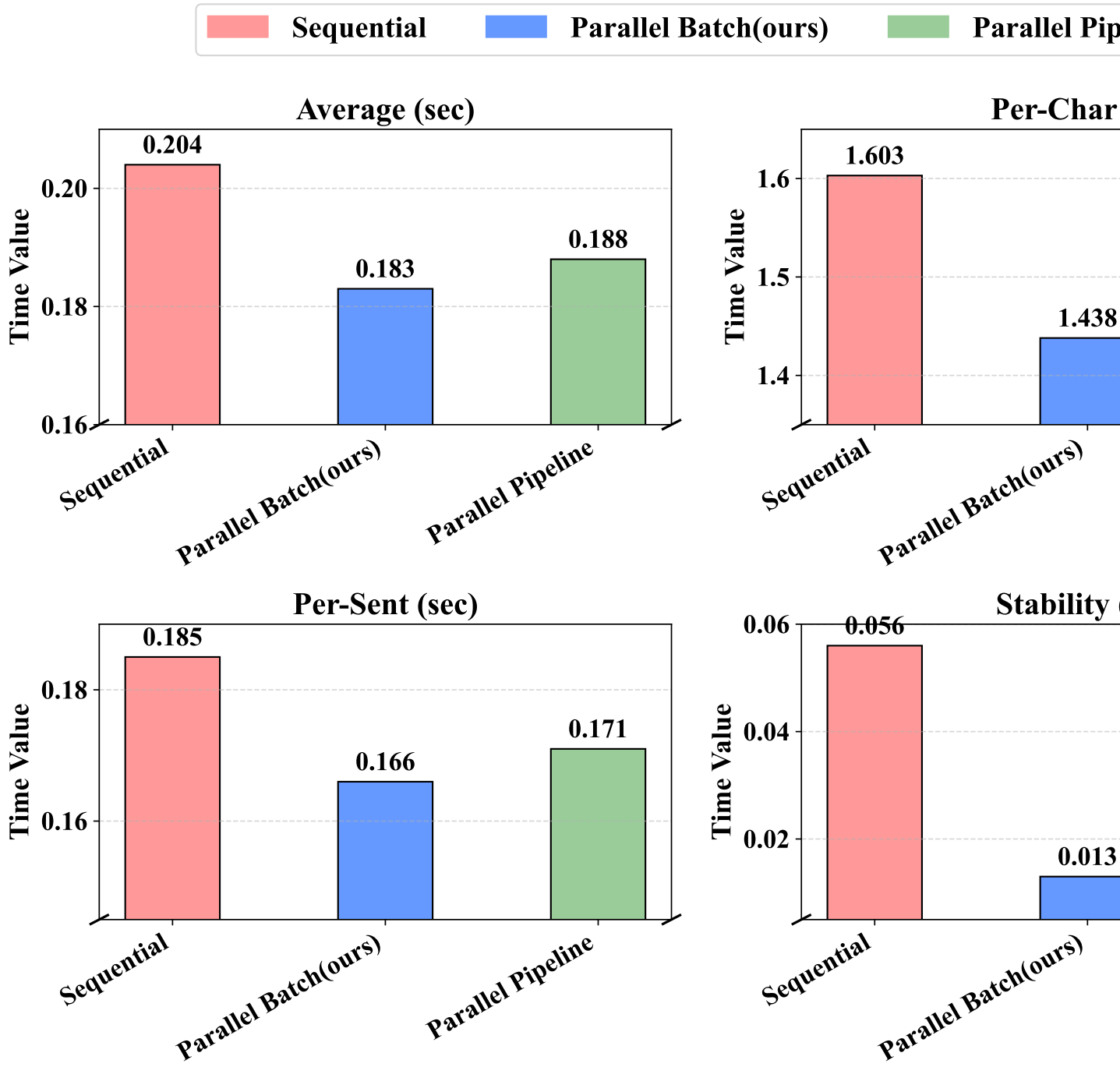


Figure 3: 通过比较 TTS 处理架构，显示出我们的并行批处理方法在速度和稳定性方面都有显著提高。

3 结论

在本文中，我们提出了一种创新的免训练 MLLM 编排框架，为自然流畅的多模态人机交互提供了创新的解决方案。与传统方法需要大量训练数据进行特征级融合不同，我们的框架通过智能编排实现了多模态能力的优雅整合。我们的实验结果展示了该框架的几个独特优势。中央控制器 LLM，凭借其出色的自然语言理解能力，能够自动分解复杂任务，并通过特定标记准确调用相应的专家模型。这种灵活的路由机制不仅能够统一处理多模态输入，还实现了基于任务特征的智能任务分解和动态调度。在处理复杂的跨模态任务时，我们创新设计了一种基于语义的批处理 TTS 机制，通过智能分割、并行处理和结果合并，显著提高了系统响应效

率。值得注意的是，我们的统一内存池系统通过标准化的标记封装，成功解决了传统系统在多模态内存管理中面临的挑战，例如模态处理时机、模型选择和跨模态交互。该设计不仅消除了在多个独立模型之间切换的不便，而且通过智能上下文管理实现了连贯且自然的多轮对话体验，为用户提供了类似于与单一超级大语言模型（LLM）互动的体验。我们框架的一个重要优势是专家模型池可以在任何时候动态扩展或减少，而无需系统重新训练，从而在适应新功能和需求时提供了极大的灵活性。我们相信，这种基于 LLM 编排的方法将为构建更智能和自然的人机交互系统开辟新的可能性。

References

[1] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[2] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge. *arXiv preprint*, 2024.

[3] Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. Llama-omni: Seamless speech interaction with large language models. *arXiv preprint arXiv:2409.06666*, 2024.

[4] Tianyu Yu, Haoye Zhang, Yuan Yao, Yunkai Dang, Da Chen, Xiaoman Lu, Ganqu Cui, Taiwen He, Zhiyuan Liu, Tat-Seng Chua, et al. Rlaif-v: Aligning mllms through open-source ai feedback for super gpt-4v trustworthiness. *arXiv preprint arXiv:2405.17220*, 2024.

[5] Chaoyou Fu, Haojia Lin, Zuwei Long, Yunhang Shen, Meng Zhao, Yifan Zhang, Shaoqi Dong, Xiong Wang, Di Yin, Long Ma, et al. Vita: Towards open-source interactive omni multimodal llm. *arXiv preprint arXiv:2408.05211*, 2024.

[6] Jiaxing Zhao, Qize Yang, Yixing Peng, Detao Bai, Shimin Yao, Boyuan Sun, Xiang Chen, Shenghao Fu, Weixuan Chen, Xihan Wei, and Liefeng Bo. Humanomni: A large vision-speech language model for human-centric video understanding. *arXiv preprint arXiv:2501.15111*, 2024.

[7] Yadong Li, Haoze Sun, Mingan Lin, Tianpeng Li, Guosheng Dong, Tao Zhang, Bowen Ding, Wei Song, Zhenglin Cheng, Yuqi Huo, et al. Baichuan-omni technical report. *arXiv preprint arXiv:2410.08565*, 2024.

[8] Haotian Liu et al. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[9] Gemini Team et al. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[10] Byung-Kwan Shin et al. X-llava: Enhanced cross-lingual large vision-language alignment. *arXiv preprint*, 2024.

[11] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[12] Aaron Grattafiori et al. The llama 3 herd of models. *IEEE Spectrum*, 2024.

[13] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[14] Xiong Wang, Yangze Li, Chaoyou Fu, Yunhang Shen, Lei Xie, Ke Li, Xing Sun, and Long Ma. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm. *arXiv preprint arXiv:2411.00774*, 2024.

[15] Zhifei Xie and Changqiao Wu. Mini-omni2: Towards open-source gpt-4o with vision, speech and duplex capabilities. *arXiv preprint arXiv:2410.11190*, 2024.

[16] Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

[17] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[18] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[19] Yifan Du, Zikang Liu, Junyi Li, and Wayne Xin Zhao. A survey of vision-language pre-trained models. *arXiv preprint arXiv:2202.10936*, 2022.

[20] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

[21] Somnath Kumar, Yash Gadhia, Tanuja Ganu, and Akshay Nambi. Mmctagent: Multi-modal critical thinking agent framework for complex visual reasoning. *arXiv preprint arXiv:2405.18358*, 2024.

[22] Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, et al. Llava-plus: Learning to use tools for creating multimodal agents. In *European Conference on Computer Vision*, pages 126–142. Springer, 2024.

[23] Wei-Ge Chen, Irina Spiridonova, Jianwei Yang, Jianfeng Gao, and Chunyuan Li. Llava-interactive: An all-in-one demo for image chat, segmentation, generation and editing. *arXiv preprint arXiv:2311.00571*, 2023.

[24] Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, Zhaozhuo Xu, and Chaoyang He. Llm multi-agent systems: Challenges and open problems. *arXiv preprint arXiv:2402.03578*, 2024.

[25] Weize Chen, Jiarui Yuan, Chen Qian, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Optima: Optimizing effectiveness and efficiency for llm-based multi-agent system. *arXiv preprint arXiv:2410.08115*, 2024.

[26] Lizhou Cao, Huadong Zhang, Chao Peng, and Jeffrey T Hansberger. Real-time multimodal interaction in virtual reality-a case study with a large virtual interface. *Multimedia Tools and Applications*, 82(16):25427–25448, 2023.

[27] Songtao Li and Hao Tang. Multimodal alignment and fusion: A survey. *arXiv preprint arXiv:2411.17040*, 2024.

[28] Zhihua Duan and Jialin Wang. Exploration of llm multi-agent application implementation based on langgraph+ crewai. *arXiv preprint arXiv:2411.18241*, 2024.

[29] Bo Qiao, Liqun Li, Xu Zhang, Shilin He, Yu Kang, Chaoyun Zhang, Fangkai Yang, Hang Dong, Jue Zhang, Lu Wang, et al. Taskweaver: A code-first agent framework. *arXiv preprint arXiv:2311.17541*, 2023.

[30] Ling Yue, Sixue Xing, Jintai Chen, and Tianfan Fu. Clinicalagent: Clinical trial multi-agent system with large language model-based reasoning. In *Proceedings of the 15th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 1–10, 2024.

[31] Jingyun Sun, Chengxiao Dai, Zhongze Luo, Yangbo Chang, and Yang Li. Lawluo: A chinese law firm co-run by llm agents. *arXiv preprint arXiv:2407.16252*, 2024.

[32] Yinzhu Quan and Zefang Liu. Invagent: A large language model based multi-agent system for inventory management in supply chains. *arXiv preprint arXiv:2407.11384*, 2024.

[33] Yoichi Ishibashi and Yoshimasa Nishimura. Self-organized agents: A llm multi-agent framework toward ultra large-scale code generation and optimization. *arXiv preprint arXiv:2404.02183*, 2024.

[34] Xuechen Liang, Meiling Tao, Yinghui Xia, Tianyu Shi, Jun Wang, and JingSong Yang. Cmat: A multi-agent collaboration tuning framework for enhancing small language models. *arXiv preprint arXiv:2404.01663*, 2024.

[35] Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Arik. Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems*, 37:132208–132237, 2024.

[36] Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

[37] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[38] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[39] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization. *Text Reading, and Beyond*, 2, 2023.

[40] Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, et al. Qwen2. 5-omni technical report. *arXiv preprint arXiv:2503.20215*, 2025.

[41] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

[42] Pan Zhang, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Rui Qian, Lin Chen, Qipeng Guo, Haodong Duan, Bin Wang, Linke Ouyang, et al. Internlm-xcomposer-2.5: A versatile large vision language model supporting long-contextual input and output. *arXiv preprint arXiv:2407.03320*, 2024.

[43] Qingpei Guo, Kaiyou Song, Zipeng Feng, Ziping Ma, Qinglong Zhang, Sirui Gao, Xuzheng Yu, Yunxiao Sun, Jingdong Chen, Ming Yang, et al. M2-omni: Advancing omni-mllm for comprehensive modality support with competitive performance. *arXiv preprint arXiv:2502.18778*, 2025.

[44] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[45] Anthropic. Claude-3.5. `https://www.anthropic.com/news/claude-3-5-sonnet`, 2024. Accessed: 2024-02-11.

[46] OpenAI. Gpt-4v. `https://openai.com/index/gpt-4v-system-card/`, 2023. Accessed: 2023-02-09.

[47] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

[48] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024.

[49] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024.

[50] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, et al. Lvbench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*, 2024.

[51] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024.

[52] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.

[53] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024.

[54] Zhibo Yang, Jun Tang, Zhaohai Li, Pengfei Wang, Jianqiang Wan, Humen Zhong, Xuejing Liu, Mingkun Yang, Peng Wang, Yuliang Liu, et al. Cc-ocr: A comprehensive and challenging ocr benchmark for evaluating large multimodal models in literacy. *arXiv preprint arXiv:2412.02210*, 2024.

[55] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 235–251. Springer, 2016.

[56] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022.

A

附录

A.1

A. MLLM 编排算法

本节介绍了我们的 MLLM Orchestration 框架的完整算法，详细说明了多模态交互处理的逐步过程，从语音输入处理到最终响应生成。

Algorithm 1 多模态学习模型编排算法

Require: User speech input s_t , Cross-modal memory M_{t-1}
Ensure: Speech response a_t

procedure ORCHESTRATIONPIPELINE(s_t, M_{t-1})
  $q_t \leftarrow f_{\text{ASR}}(s_t)$ ▷ Speech recognition
  $Y_t \leftarrow f_{\text{ctrl}}(q_t, M_{t-1})$ ▷ Controller LLM reasoning
  $C_t, S_t \leftarrow \text{SplitTokens}(Y_t)$ ▷ Split content and control tokens
  if $[S.stop] \in S_t$ **then**
    CLEARTTSQUEUE
    $M_t \leftarrow M_{t-1} \cup \{q_t, \text{interrupt\_flag}\}$
    return
  if $[S.listen] \in S_t$ **then**
    $M_t \leftarrow M_{t-1} \cup \{q_t, \text{await\_completion}\}$
    return
  $\mathcal{M} \leftarrow \delta_{\text{modality}}(S_t)$ ▷ Modality selection
  $D_t \leftarrow \emptyset$ ▷ Initialize multimodal data collection
  for $m_i \in \mathcal{M}$ **do**
    if $m_i = \text{vision}$ **then**
      $v_t \leftarrow h_{\text{vision}}(q_t, M_{t-1})$
      $D_t \leftarrow D_t \cup \{v_t\}$
    else if $m_i = \text{reasoning}$ **then**
      $r_t \leftarrow h_{\text{reasoning}}(q_t, M_{t-1})$
      $D_t \leftarrow D_t \cup \{r_t\}$
  $\widetilde{Y}_t \leftarrow I(Y_t, D_t)$ ▷ Multimodal response fusion
  $T \leftarrow \text{segment}(\widetilde{Y}_t)$ ▷ Semantic segmentation, see Appendix E
  for $i \leftarrow 1$ to $|T|$ parallel **do**
    $a_{t,i} \leftarrow g(T_i)$ ▷ Parallel TTS synthesis
  $a_t \leftarrow \text{stream\_concat}(a_{t,1}, \ldots, a_{t,|T|})$ ▷ Stream concatenation
  $M_t \leftarrow h_{\text{compress}}(M_{t-1} \cup \{q_t, D_t, \widetilde{Y}_t\})$ ▷ Memory compression
  return a_t
function δ_{MODALITY} (S_t)
  return $\{m_i | [S.need\_m_i] \in S_t, m_i \in \mathcal{M}\}$ ▷ $\mathcal{M}$ is set of available modalities
function SPLITTOKENS(Y_t)
  $C_t \leftarrow \{y_i | y_i \in Y_t, y_i \text{ is content token}\}$
  $S_t \leftarrow \{y_i | y_i \in Y_t, y_i \text{ is control token}\}$
  return C_t, S_t

A.2

B. 核心指令集

本节详细介绍了 MLLM 编排框架的核心指令集，作为控制器 LLM 和专业化模型之间的基本通信协议。该指令集设计有三个主要目标：

- 语义清晰：每个指令都有明确的意义和可预测的行为
- 动态协调：实现跨异构模式的实时任务路由
- 架构可扩展性：通过标准化模式支持新模态的无缝集成

Table 4: 多模态交互控制的核心指令集

Instruction	Functionality and Usage Context
`[S.stop]`	Immediate process termination and resource release
`[S.need_vision]`	Activates visual processing pipeline (object detection, scene parsing)
`[S.need_reasoning]`	Invokes logical inference and causal analysis modules
`[S.speak]`	Generates verbal response with prosody control
`[S.listen]`	Enters passive listening mode for continuous input
`[S.need_*]`	Extensible pattern for future modality integration

A.3

C. 核心提示设计

该提示体系结构通过三个关键设计原则实现自然的人机交互：

- 角色专门化：系统责任的明确分工
- 响应优化：平衡信息密度和对话流程
- 模态路由：基于输入分析的智能资源分配

```
Controller LLM prompt : You are an AI assistant responsible for coordinating specialized
models. Your primary functions include:
(1) 通过多模态输入分析用户意图。
        如果用户输入不完整或需要进一步澄清，则输出特殊控制符号 [S.listen] 。
        如果用户打算中断，则输出特殊控制标记 [S.stop] 。
        当需要您的回应时，输出特殊控制符号 [S.speak] 。
(2) 动态地将任务路由到适当的专家模型。
        使用 [S.need_vision] 进行视觉查询。
        使用 [S.need_reasoning] 进行分析性任务。
(3) 所有回答必须遵循“特殊控制标记 + 回应内容”的格式；其他格式的回应将被拒绝。
```

A.4

D. 内存池方案

跨模态记忆系统采用具有以下特征的基于 JSON 的模式：

- 结构完整性：跨模态的统一数据表示
- 检索效率：为低延迟访问优化的索引
- 上下文保持：长期对话状态维护

关键架构特征：

- 模块化设计支持可插拔扩展
- 内容感知压缩算法
- 时间和语义索引层
- 版本控制的内存修订

```
Memory JSON:
{
    "id": "uuid4",
    "modality": "vision|audio|text|reasoning",
    "content": {
        "type": "mime_type",
        "data": "encrypted_blob",
        "embedding": "float_vector",
        "metadata": {
            "source": "input_device",
            "timestamp": "iso8601",
            "context": "semantic_tags"
        }
    },
    "turn_id": "sequential_counter",
    "references": ["related_uuids"],
    "priority": "0.0-1.0",
    "compression": {
        "algorithm": "zstd|lz4",
        "ratio": "float"
    }
}
```

A.5

E. TTS 分割规则

该文本到语音模块采用基于规则的分段策略来优化语音合成的质量和时延。分段规则旨在保持语义一致性的同时，实现并行处理。以下是详细的分段规则：

```
Rule 1:  Natural punctuation boundaries —e.g., periods, commas, semicolons
Rule 2:  Discourse markers and conjunctions —e.g., "however", "therefore", "and"
Rule 3:  Syntactic boundaries —e.g., subject-predicate splits, subordinate clauses
```

每个片段通常包含 7 到 15 个单词，以平衡合成质量和并行化效率。这个粒度既确保了语义完整性和自然的重音模式，同时也保持了计算效率。例如，句子“The cat sat on the mat, while the dog slept peacefully”会被划分为两个片段：“The cat sat on the mat”和“while the dog slept peacefully”。