借助大语言模型的语音识别，实现对软件的口头控制。应用于沉浸式分子图形学的增强现实/虚拟现实 Web 应用程序。

Fabio Cortés Rodríguez 和 *Luciano Abriata*

瑞士洛桑联邦理工学院

摘要

本项目成功开发、评估并将语音用户界面（VUI）集成到我们正在开发的一款用于沉浸式分子图形学的 Web 应用程序中。该应用程序提供增强现实/虚拟现实（AR/VR）环境，用户可以用双手操控分子，但这意味着无法通过常规的鼠标和键盘图形用户界面（GUI）来控制应用程序。本项目开发的语音 VUI 系统解决了这一问题，让用户能够通过自然的口头（或输入）指令轻松控制应用程序。为实现这一 VUI，我们评估了两种不同的自动语音识别（ASR）系统：Chrome 原生语音 API 和 OpenAI 的 Whisper v3。尽管 Whisper 的浏览器兼容性更广，但它容易对专业科学术语产生“幻觉”，这一点非常棘手。因此，我们选择了 Chrome 的自动语音识别，因其稳定、快速且可靠。为了将转录后的语音转换为软件指令，我们测试了两种由大语言模型（LLM）驱动的方法：生成可执行代码，或调用预定义函数。最终采用了由 OpenAI 的 GPT-4o-mini 驱动的函数调用方法，因为与更复杂且容易出错的代码生成方法相比，它更安全、高效、可靠。最终的 VUI 将 Chrome 的自动语音识别与基于大语言模型的函数调用模块相结合，使用户能够使用自然语言控制应用程序，具体示例见本报告中的视频链接。我们提供了在线示例链接，展示所有中间组件，并详细介绍了如何设计大语言模型的提示词，以教会它调用函数，以及如何清理转录后的语音并让模型在生成函数调用时进行解释。为更好地展示最终系统，我们还提供了一个视频示例。

引言

近期研究探索了将自动语音识别与大语言模型结合，通过自适应边缘云编排，将语音输入转换为可供下游系统执行的命令。[1] 在工业环境中，研究表明，采用大语言模型增强的语音用户界面（VUI，与图形用户界面 GUI 相对）能够缓解传统界面的局限，尤其突显了其在控制任务中的灵活性和自然语言落地能力。[2,3] 本研究的核心假设是，同样地，语音用户界面在增强现实与虚拟现实场景中也将大有用处。

本文报告了一个系统的开发与评估，该系统旨在将基于大语言模型（LLM）的语音识别（自动语音识别，ASR）和自然语言处理（NLP）能力，集成到使用增强现实/虚拟现实（AR/VR）的分子图形学与建模软件中。本研究服务于我们的总体目标：通过在增强现实/虚拟现实工具中启用基于语音的命令[4–7]，同时保持其在不同设备和平台上的灵活性与功能，增强用户与软件的交互。更具体地说，在本项工作中，我们探讨了如何创建一个可插入特定

完善的网络应用程序。该网络应用程序用于多用户增强现实和虚拟现实环境中的分子可视化与建模。作为本项目的容器，这类网络应用程序非常合适，因为用户在增强现实/虚拟现实会话中通常需要使用双手与虚拟物体进行交互，因此，采用基于语音的系统与设备和软件进行交互，可能会极其有用。

本报告概述了整个工作过程中采用的方法、所评估的不同自动语音识别系统和自然语言处理组件的性能、它们与我们的分子图形学与建模增强现实/虚拟现实网络应用中其他组件的集成情况，以及其潜在的更广泛影响；同时也讨论了所面临的挑战和可能的改进方向。在全文中，我们提供了实现部分测试的网页链接，以及记录自动语音识别、自然语言处理和ASR-NLP流水线示例输出的截图。

第1节：自动语音识别（ASR）（或文本转语音，TTS）系统

在本项目期间，我们测试了两种不同的自动语音识别系统：Chrome 的原生语音识别系统，该系统通过 Speech API 提供，而 Chrome 是少数支持该 API 的浏览器之一；以及 Whisper v3，这是 OpenAI 的旗舰自动语音识别模型，建立在音频词元识别与语言处理相结合的基础上。Chrome 的 Speech API 原生集成于其面向智能手机、平板电脑和计算机的浏览器版本中，因此极易使用和编写代码。此外，它还可以零成本使用。Whisper v3 是开源的，这意味着原则上也可以免费安装和部署，但在实际场景中，以可通过网络访问的形式部署它并不简单，当然也无法免费实现。为避免相关工作和成本，我们选择通过一家名为 Gladia 的公司提供的 API 使用 Whisper v3；该公司为我们提供了免费额度，以便开展各项测试。

Chrome 的自动语音识别系统可在浏览器内原生运行，而其他支持 Speech API 的浏览器并不多。这导致应用依赖特定浏览器，限制了它在沉浸式环境中的应用，尤其是在不支持 Chrome 的环境中。尽管有适用于 AR/VR 头显的基于 Chromium 的网络浏览器，但它们并不支持 Speech API。另一方面，通过 Gladia 的 API 使用 Whisper v3，无需处理运行模型的复杂性，还能借助相应的基础设施确保响应迅速、运行高效，成本较低（但并非零成本）。值得一提的是，就我们的工作而言，Gladia 不仅提供了免费的测试额度，还提供了足够的 JavaScript 代码片段，帮助我们通过套接字将其集成到网络应用中。

在我们自己的测试中，Whisper 方案展现出预期的兼容性，可在笔记本电脑和智能手机上的所有受测浏览器中运行，包括 Chrome、Brave、Firefox 和 Safari。关键的是，它还可在 Meta Quest 2 和 3 等 AR/VR 头显上的 Meta Quest 浏览器中运行，从而能够在沉浸式环境中直接进行语音交互。这一能力对我们的应用是一大优势，因为分子图形工具旨在 AR/VR 环境中运行，在这些环境下，传统界面可能并不实用；不过，Meta Quest 浏览器并不支持 Speech API。

最后，需要 acknowledge 的是，除了识别准确率之外，近期研究还评估了不同自动语音识别系统的能耗和环境成本；这对于已部署的语音用户界面而言，是一项实际考量。[8]

首次测试两种自动语音识别系统所用的示例代码和网页
Chrome 和 Whisper 自动语音识别系统的最基础实现可通过以下链接获取：

- http://lucianoabriata.altervista.org/VUIsProject/basic-deploy---Chrome-STT.htm l

- http://lucianoabriata.altervista.org/VUIsProject/basic-deploy---Whisper-STT.html

注意：Whisper 应用程序需要 Gladia API 密钥。

以下两个示例采用了定制的语法扩展，并测试了这种方法能否克服自动语音识别系统在检测和转录高度专业术语时的局限：

- https://lucianoabriata.altervista.org/VUIsProject/grammar-expansion---Chrome-STT.html（仅适用于 Chrome）
- https://lucianoabriata.altervista.org/VUIsProject/grammar-expansion---Whisper-STT.html

性能观察

就性能而言，Chrome 的自动语音识别系统存在一些可靠性问题。持续聆听较长时间后，它偶尔会停止运行。不过，在网络应用的代码中关键位置以编程方式重启自动语音识别系统，可以相当可靠地缓解这一问题。除这一缺点外，Chrome 的自动语音识别功能表现稳定，速度也略快于 Whisper。

尽管 Whisper 在设备和浏览器兼容性方面更加灵活，但它也带来了挑战，主要是会产生幻觉，也就是持续输出与实际输入无关的词元。当系统收到语法或专业术语方面的提示时，这一问题尤为明显。这是个重大问题，因为正如原始项目提案中所述，我们计划通过语法注入来提高系统对分子科学相关词汇的识别能力，而这些词汇在日常对话中并不常见。例如，Whisper 在转录领域专用术语时，有时会生成无意义或非字母数字的输出，使下游清理和解读变得困难。尽管 Whisper 在英语、西班牙语、法语和意大利语的一般语音识别方面表现相对出色，但它在处理专业词汇时遇到的困难，可能是由于训练数据中此类术语的代表性不足。最终，我们发现，这一局限妨碍了它在分子建模领域高度专业化术语方面的应用。

选择一种自动语音识别系统

最终，我们选择了 Chrome 的自动语音识别系统，因为它更加稳定、成本更低且运行速度更快。虽然这一选择牺牲了对非 Chrome 浏览器的兼容性，尤其是在 AR/VR 头显上，但考虑到项目的迫切需求以及该系统极其简单的实现方式，这种取舍是可以接受的。

第 2 节：语音到命令转换

测试的方法

按照原始提案中的概述，我们针对不断扩展的分子图形工具，在大语言模型的上下文中编写并测试了两种方法，将用户命令转化为操作。第一种方法是调用预定义函数，第二种则是根据转录文本生成可执行代码。如果运行正常，后一种方法预计会灵活得多，但这会带来安全隐患，因为我们无法排除生成并执行恶意代码的可能性。此外，我们最终还发现了其他问题，详见下文。

相应地，第一种方法（调用预定义函数）缺乏灵活性，除非有足够数量的函数被硬编码并预先定义；但其优势在于，这种方法运行更快，消耗的大语言模型令牌更少，并且不会带来安全问题，因为没有代码

被执行—代码中已存在的预定义函数只需在大语言模型的输出 中被检测到，然后进行解析并调用。这种方法依赖于为 OpenAI的 GPT-4o-mini精心设计的提示词；GPT-4o-mini是我们项目所选用的大语言模型，因为它速度快且成本低廉。经过多次迭代，提示词得到了优化，能够准确地将转录文本映射到特定函数，从而确保性能一致。这种方法被证明非常有效，在某些情况下甚至能够纠正转录错误。通过向提示词中提供定义明确的示例，系统能够可靠地理解用户命令并执行相应操作。此外，我们观察到，提供解释每个函数为何以及如何被在提示词示例中调用的注释，能够带来更好的生成结果。

第二种方法，即通过生成可执行脚本来实现的方法，展现出了很大的灵活性潜力。事实上，令人惊讶的是，只要有充足的时间和词元，在工具内生成并执行代码的方法就会非常强大且前景广阔，前提是能够解决安全问题。不过，测试该方法时，我们发现了一些超出预期的挑战，尤其是安全风险方面。首先，提示词要长得多（因为它们不仅需要展示函数调用，还必须为每项预期任务提供多行代码），而且输出所需的词元数量也更多，这增加了复杂性、拖慢了执行速度并推高了成本。例如，与预定义的函数调用相比，这种方法所需的词元数量是其五到十倍（因为它往往需要生成多行代码，而第一种方法只需几个函数调用）。其次，我们观察到，提示词过长时，出错的概率会大幅上升；在某些情况下，大语言模型的输出甚至会被截断，而这些情况都会对程序执行造成致命后果。

最终实现

考虑到上述挑战，我们采用预定义函数方案作为语音到命令的转换的主要方法。这种方法使我们能够兼顾可靠性和效率，同时保留了扩展工具功能的高度灵活性。每当工具新增一项功能，我们都会在提示词中加入相应示例，确保系统能够无缝处理扩展后的功能。

以下摘录自我们精心编写的核心提示词（截至本报告提交时），用于教大语言模型如何在我们的应用程序中调用函数；后面附有前几个输入输出对示例：

{ role: "system", content: "你会收到语音识别文本，并根据示例将其转换为命令。如果收到的文本不是英语，你只需先将其翻译成英语，然后执行。如果你不理解请求，就运行命令 didntUnderstand()。你只能输出以下函数：countAtoms()、acknowledge()、sayTime()、sayDate()、zoomIn()、zoomOut()、changeTemperature()、changeUpdateRate()、setTemperature()、startSimulation()、stopSimulation()、writePDB()、select()、changeUpdateRate()、speakUp()、spacefill、sticks、color 和 didntUnderstand()。对于没有相近示例的请求，要进行智能转换，例如，“cartoon”指的是使用主链选择，然后应用“spacefill 1; sticks 2”。确保不要使用不存在的内容，例如 color byref。对于没有相近示例的请求，要进行智能转换，例如，“cartoon”指的是组成主链的原子，比如蛋白质中的 N、C 和 CA 原子，或核酸中的 P 原子。请注意，由于语音识别错误，提供的输入中有些词可能有误，例如，“some”可能需要替换为“zoom”，“so mean”可能是“zoom in”，“change”实际上可能是“chain”或“chains”，“handball”实际上是“HandMol”，“mini mice”实际上是“minimize”，等等。最后，请确保像示例中那样输出注释，说明调用每个函数的原因。"

{ role: "user", content: "告诉我原子的数量" },
{ role: "assistant", content: " //使用 countAtoms()\ncountAtoms();" },

{ role: "user", content: "升高温度" },
{ role: "assistant", content: " //有一个专门的命令可以按指定数值调整温度，另一个命令则可以将温度设为指定值。此请求要求升高温度，因此我们使用 changeTemperature()，并传入一个正数作为参数\nchangeTemperature(+30);" },

{ role: "user", content: " 将所有带正电荷的残基显示为球体。" },

{ role: "assistant", content: "//只需 select 带正电荷的氨基酸，即精氨酸和赖氨酸，并使用 'spacefill 3' 将它们显示为半径为 3 的球体\nselect('resname ARG LYS'); spacefill 3;" },

{ role: "user", content: "停止模拟" }, { role: "assistant", content: "//用户要求停止（暂停）模拟\nstopSimulation();"}, { role: "user", content: "写出坐标" }, { role: "assistant", content: "//调用 writePDB() 函数\nwritePDB();" }, { role: "user", content: "突出显示 1 号残基" },

{ role: "assistant", content: "//要突出显示一个原子，我们需要增大它的球体，因此这里使用 'spacefill 3'\nselect('resid 1'); spacefill 3; sticks 1;" },

{ role: "user", content: "不，等等，把它显示成红色" },

{ role: "assistant", content: "//我们先重新选择原子，然后应用颜色\nselect('resid 1'); color red;" },

{ role: "user", content: "按电荷为 C 链着色" },

{ role: "assistant", content: "//具体选择 C 链，然后应用氨基酸电荷的标准颜色\nselect('chain C'); color white; select('resname ASP GLU and chain C'); color red; select('resname ARG LYS and chain C'); color blue;",

{ role: "user", content: "把球体调大" },

{ role: "assistant", content: "//未指定特定的原子或分子，因此我们选择全部并增大半径\nselect('all'); spacefill 3;" },

{ role: "user", content: "再大很多" },

{ role: "assistant", content: "select('all'); spacefill 6;" },

},

...

这里有几点相关说明。首先，上面的摘录只展示了具有代表性的用户/助手对示例。截至本报告提交之日，完整提示词约有 300 行代码，共约 9,000 个词元。其次，所选示例展示了我们纳入的一些功能，既包括应用特定层面的功能（控制模拟、切换视图和颜色、请求系统相关信息等），也包括通用的语音用户界面功能，例如根据上下文更好地理解请求的能力（例如，提示词末尾附近先是“让球体变大”，接着是“再大得多”）。第三，请注意，许多示例输出都包含一条 JavaScript 格式的注释，也就是以两个斜杠（//）开头；我们发现这样能让输出更好，这可能是因为它们能帮助大语言模型更仔细地“思考”用户的请求。

我们提供了这些网页，用于测试 Chrome 和 Whisper 自动语音识别系统的代码生成和函数调用（需要 API 密钥）：

- https://lucianoabriata.altervista.org/VUIsProject/request-to-code---Chrome-STT.html(仅适用于 Chrome)
- https://lucianoabriata.altervista.org/VUIsProject/request-to-function-call---Chrome-STT.html (仅适用于 Chrome)
- https://lucianoabriata.altervista.org/VUIsProject/request-to-function-call---Whisper-STT.html

为了获得完整体验，读者可参阅下一节所示的示例视频和屏幕截图。

第 3 节：融入我们不断发展的增强现实/虚拟现实分子图形学工具 I

在提交本报告前的最后几个月里，我们成功将如上所示、使用 Chrome 的自动语音识别并结合 GPT-4o-mini 进行转录到函数的转换的流程，集成到增强现实/虚拟现实分子图形学网络应用的原型草案中。该应用程序本身仍需大量完善，但得益于本资助项目使我们能够投入时间，语音用户界面已经相当完整。

截至本报告提交之日，语音用户界面系统已能够将以自然语言说出（或键入）的输入转换为操作，从而高效、无缝地控制网络应用。因此，用户可以例如要求系统“将温度提高 50 度”，网络应用便会相应提高分子模拟的温度。同样，用户也可以要求系统启动或停止模拟、放大或缩小、切换可视化效果等。

此外，我们针对 GPT-4o-mini 优化的 大语言模型提示词，使其能够进行“智能”单位换算、清理转录错误，并将英语以外的语言翻译成英语（自动语音识别模块仅转录英语请求，但以任何语言键入请求均可）。这些积极方面并不令人意外，因为比较研究表明，在某些低资源和语码转换场景中，基于大语言模型的自动语音识别能够胜过通用模型，这凸显了语境语言理解的重要性。[9,10]

而当我们把用户请求和大语言模型的输出追加到聊天式历史记录时，系统便能跟踪整段对话；于是，用户例如可以说“把所有原子color成红色”，接着说“不对，你最好按原子来color它们”，系统就会理解这两项请求。这最后一项特定功能对于流畅、自然的交流十分重要，它是通过大语言模型提示词的输入输出对中如下这类行来实现的：

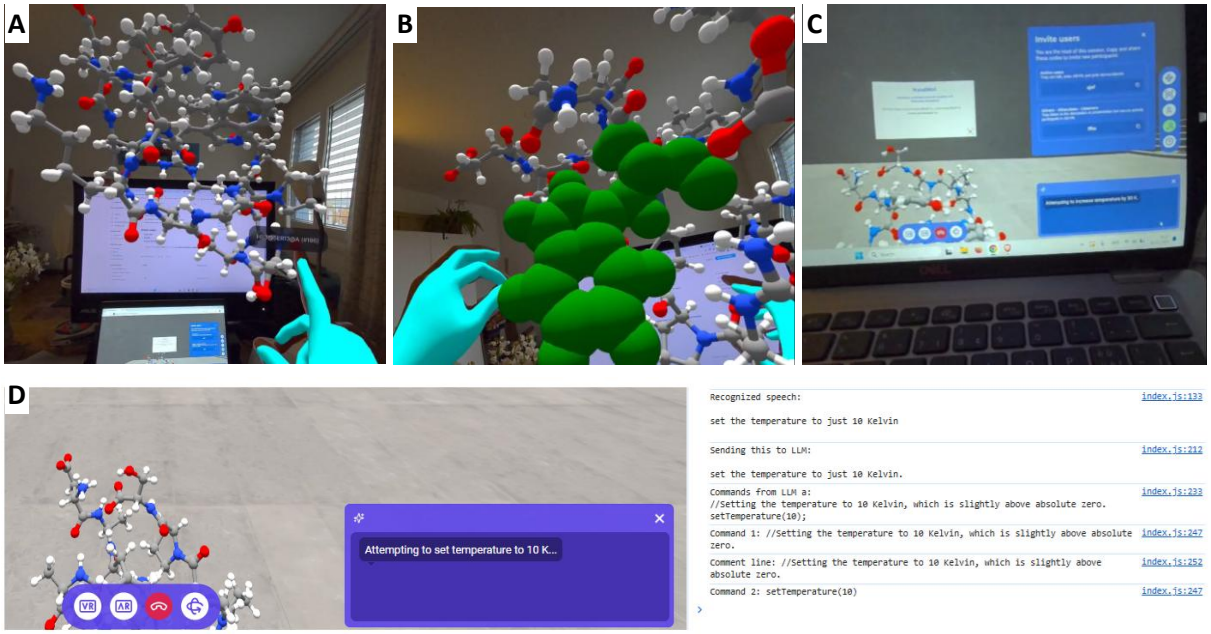
```
{ role: "user", content: "提高温度" }, { role: "assistant", content: "
changeTemperature(+30);" }, { role: "user", content: "再次" }, { role: "
assistant", content: "changeTemperature(+30);" },
```

能力与局限

要亲眼看到它的实际效果并从而掌握其能力，最好的方式当然是通过真实示例。因此，我们录制了一段视频，展示一个将此处开发的自动语音识别系统与我们不断发展的分子图形学工具集成在一起的原型，拍摄于一台以增强现实模式运行我们网络应用的 Meta Quest 3 设备内部（视频演示见https://www.youtube.com/watch?v=DozF99mW_KU，示例截图见图下方）。请注意，由于我们所选的自动语音识别系统仅在 Chrome 中运行，我们不得不在一台管理会话的计算机上运行它（这得益于该工具的多用户特性）。

当然，这种方法也有局限性，我们观察到的问题主要在于处理嘈杂环境和非标准口音时。 不过，受测的两个自动语音识别系统都存在这一情况：它们在安静、没有背景噪音的环境中表现最佳（此外，Whisper 还受到前文所述问题的影响）。

初步可用性测试显示，用户对此反响积极，尤其认可界面的直观易用性。不过，我们还需要进行更深入的测试；只有等到应用程序更加完善，用户能够运行沉浸式交互式分子图形与建模，并在真实场景中确实需要语音用户界面来处理请求时，才有可能开展这类测试。



以大语言模型为核心、基于 Chrome 的语音用户界面，已以草稿形式集成到我们的分子图形应用中，以下为其截图。（A-C） 从 AR/VR头显内部拍摄的演示视频截图，视频见 https://www.youtube.com/watch?v=DozF99mW_KU。（A）和（B）展示了基本的沉浸式功能，以及用户如何与分子图形交互以获取信息或指示信息（A），并与分子动力学模拟交互以执行操作（B）。这两种情况下，后方屏幕都显示主机用户一侧的界面，并在 Chrome 浏览器中运行，以使用其内置的语音识别系统。（C）中，视图聚焦于人工智能/大语言模型语音用户界面（右下方的蓝色框），它正在执行用户要求提高模拟温度后的指令。（D）为网页浏览器截图，其中主机正在运行语音识别；画面聚焦于语音用户界面/人工智能模块，并显示控制台日志，以展示语音如何先被转录，然后转录文本如何由大语言模型转换为函数调用和注释。

第 4 节：总体影响与更广泛的影响

这个项目让我们在将自动语音识别和自然语言处理系统集成到 Web 应用程序方面积累了大量宝贵经验。学习如何优化提示词，将文本转换为函数调用，也非常有价值，因为这为分子图形学与建模以外领域的广泛应用带来了许多可能性。一些此类应用已经在筹备中；例如，一家开发基因组学和转录组学数据分析软件的团队看到我们的 ASR-NLP流水线运行后，便联系了我们，询问能否帮助他们开发一个模块，采用相同的方法，让用户通过自然语言请求处理和分析基因组学及转录组学数据。这与我们成果中最扎实的部分有关，也就是将输入或转录的文本转换为函数调用或代码生成。因此，我们认为能够帮助这支团队实现这一令人振奋的应用；这也将展示本文所提方法的强大能力，并将其应用于一个关注度更高的领域，因为这类数据分析对基础生物学和医学具有重要意义。

最后，为了让您资助的这项工作成果也能惠及他人，我们已将所有结果、代码和提示词免费发布在网上，网址见报告各处（但需要 API 密钥）。待本报告获批后，我们还会以博客文章和/或预印本的形式发布，并采用开放获取方式。

第 5 节：挑战与未来方向

本项目期间遇到了一些挑战，报告中已详细说明，这里简要回顾一下。首先，无法扩展 Chrome 的自动语音识别语法来提高对领域专用术语的识别能力，因此其识别专业命令的准确度受到限制。不过，我们可以通过指示下游大语言模型组件检测并纠正可能的拼写错误，克服其中一些问题。例如，请参阅第 2 节“最终实现”下提供的提示词中的系统部分，其中提到：“请注意，由于语音识别错误，输入中的某些词可能有误，例如，'some' 可能需要替换为 'zoom'，'so mean' 可能是 'zoom in'，'change' 实际上可能是 'chain' 或 'chains'。”

Whisper 的幻觉倾向令人非常失望，但未来的基于人工智能的自动语音识别系统有可能优于它，也优于 Chrome 内置的自动语音识别系统。

同样，鉴于代码生成方法具有更强的能力，自动语音识别转行动的代码生成方法颇具吸引力；因此，开展更谨慎、更专门的研究或许值得一试——尽管我们已经证实了这一方法存在若干缺点，甚至发现了一些新的问题，详情见第 2 节“已测试的方法”。

报告中我们未涉及的一个方面，是网络应用以口头方式回复的可能性。虽然这并不在最初提出的工作范围内，但我们确实曾简要测试 Chrome 自带的语音合成 API，以及名为 Talkify 的第三方 API；后者为我们提供了免费额度用于测试。我们发现两者的表现都非常好，实际上优于任何系统的语音识别。因此，在项目接近结束时，我们开始在程序中加入一些语音合成功能，以实现更流畅的界面（第 3 节“功能与局限”中提供的视频链接展示了该应用，视频中还包含了一些生成音频的示例）。

后续工作将聚焦于进一步优化大语言模型提示词，并解决系统在不同条件下的稳健性问题。当我们进一步推进分子图形与建模应用程序时，尤其是开始与真实用户进行测试时，我们将不得不面对这一问题。

结论

总体而言，我们将本项目视为成功，因为我们观察到，Chrome 的自动语音识别与 GPT-4o-mini 在转换为函数调用时对转录的清理配合得相当好，为我们提供了一条清晰路径，用以实现对网络应用灵活的自然语言语音控制。正如任何出色完成的工作一样，识别局限性十分重要，这样我们才能恰当地告知用户应有何种预期，以及如何以自然语言与该工具进行最佳交互。

我们设想，如果分子图形学和建模应用程序的开发能够继续保持这样的进展，语音用户界面将使该应用程序独具特色，同时也可能促进未来用于控制其他软件的语音用户界面的发展。

参考文献

1. 用于语音到动作系统的自适应边缘云推理：采用自动语音识别和大语言模型（ASTA）。https://arxiv.org/html/2512.12769v1。2. Mukherjee, A. 等。基于大语言模型的语音用户界面：实现用户与工业机器之间的语音对话。Procedia CIRP 134, 378–383（2025）。3. Wagner, D.、Churchill, A.、Sigtia, S. 和 Marchi, E. SELMA：用于虚拟助手交互的语音交互语言模型。预印本, https://doi.org/10.48550/arXiv.2501.19377（2025）。4. Rodriguez, F. C.、Krapp, L.、Dal Peraro, M. 和 Abriata, L. HandMol：结合 WebXR、人工智能和人机交互技术，实现沉浸式、自然、协作且包容的分子建模。（2023）。5. Abriata, L. A. MolecularWeb 宇宙：基于 Web 的沉浸式多用户分子图形与建模，用于化学、结构生物学和材料科学领域的教育与工作。预印本, https://doi.org/10.48550/arXiv.2509.04056（2025）。6. Cortés Rodríguez, F. J. 等。MolecularWebXR：通过沉浸式且包容的增强现实与虚拟现实开展化学和生物学领域的多用户讨论。Journal of Molecular Graphics and Modelling 135，108932（2025）。

7. Rodríguez, F. C. 等。MoleculARweb：一个利用消费级设备开箱即用的交互式增强现实开展化学和结构生物学教育的网站。Journal of Chemical Education 98, 2243–2255 (2021)。

8. El Bahri, J., Kouissi, M. & Achkari Begdouri, M. 自动语音识别系统中能源消耗与碳足迹的比较分析：比较 Whisper 与 Google Speech-to-Text 的案例研究。Comput. Sci. Math. Forum 10, 6 (2025).

9. 低资源和语码转换场景下基于大语言模型的自动语音识别与 Whisper 的比较研究。https://arxiv.org/html/2412.00721v1.

10. Mahmood, A., Wang, J., Yao, B., Wang, D. & Huang, C.-M. 与大语言模型驱动的语音助手对话时的用户交互模式与交互故障。《国际人机交互研究杂志》195, 103406（2025）。