|  |  |
| --- | --- |
|  | **第十七届全国大学生软件创新大赛** |
| **文档编号：SWC2024-Hi安卓** |



**按键精灵**

**AutoDroid**

**技术研究报告**

**Version: 1.0.0**



**Hi安卓**

**2024-03-19**

**All Rights Reserved**

目录

[1 问题聚焦 1](#_Toc161739473)

[1.1 问题描述 1](#_Toc161739474)

[1.2 问题抽象 2](#_Toc161739475)

[1.3 问题定位 2](#_Toc161739476)

[1.4 问题评估 2](#_Toc161739477)

[1.5 问题分解 3](#_Toc161739478)

[2 相关工作 5](#_Toc161739479)

[3 技术方案 6](#_Toc161739480)

[3.1 技术方向 6](#_Toc161739481)

[3.2 技术选择 6](#_Toc161739482)

[3.3 结果期望 6](#_Toc161739483)

[4 技术实践 8](#_Toc161739484)

[4.1 使用的开发框架及依赖的库 8](#_Toc161739485)

[4.2 技术实践过程 9](#_Toc161739486)

[5 结果验证 11](#_Toc161739487)

文档修订历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **修订原因** | **版本号** | **作者** | **修订日期** | **备注** |
| **1** | **初稿** | **0.5.0** | **Hi安卓队** | **2024.01.07** |  |
| **2** | **确定研究范围** | **0.6.0** | **Hi安卓队** | **2024.01.28** |  |
| **3** | **对技术方案进行完善** | **0.8.0** | **Hi安卓队** | **2024.02.27** |  |
| **4** | **对技术实践过程进行了完善** | **0.9.0** | **Hi安卓队** | **2024.03.06** |  |
| **5** | **对结果进行了验证** | **1.0.0** | **Hi安卓队** | **2024.03.14** |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 问题聚焦

## 问题描述

如今智能手机市场进入了存量时代，各家都在寻求新的创新机遇，国产手机今年在AI领域进行了密集的布局，也成为推动国产AI进步的强助力。根据IDC 预测，到2026年，中国市场近50%的终端设备处理器将带有AI技术。后续随着AI大模型与硬件的结合陆续完善，消费电子产业链也将迎来新的创新机遇。智能手机、PC、智能眼镜、智能音箱等终端也将成为人工智能的硬件入口，创建新的生态，带动消费电子产业链的新一轮创新。在这种情况下，智能手机助手所面临的问题包括以下几个方面：

1. 界面操作复杂性与用户熟悉度不足：

对于新用户或不常使用某些应用的用户来说，面对繁杂的操作流程或不直观的用户界面(UI)，完成简单任务都可能变得困难。例如，一个购物应用可能要求用户经过多个步骤才能完成购买，或者一个社交应用的隐私设置可能隐藏在多层菜单之下。这种复杂性不仅增加了用户的认知负担，也大大降低了操作的效率。

1. 针对残障人士的可访问性挑战：

残障人士，尤其是视障用户，使用触控屏幕设备和导航复杂的移动应用界面时面临巨大挑战。传统的触控操作和基于视觉的界面设计不满足他们的需要，导致这部分用户无法充分利用智能手机应用提供的服务和功能，增强他们的日常生活质量。

1. 隐私保护与数据安全问题：

在解析用户指令和执行自动化任务的过程中，需要处理敏感的用户数据和隐私信息。如何确保这些信息在自动化过程中得到妥善处理，避免泄露给第三方或上传到云端，是项目实施中必须严格考虑的问题。用户对于隐私保护的担忧可能会阻碍他们使用自动化助手，尤其是当涉及到需要访问个人信息的任务时。

1. 跨应用操作的复杂性：

用户经常需要在不同的应用程序之间完成任务，例如，从电子邮件中复制会议链接到日历应用中创建事件。这类跨应用操作涉及不同界面和操作逻辑的切换，对于自动化系统来说是一个极大的挑战。不仅需要精确理解每个应用的UI和功能，还需要在不同应用间高效准确地传递信息和执行操作，这对于设计高度灵活和智能的自动化助手提出了更高要求。

综上本项目要解决的问题就是对以上四个方面一种实践探索。

## 问题抽象

依据1.1中的问题描述，本项目要解决的问题可以归结为：如何更好地理解用户的意图和任务的上下文，如何将与应用程序相关的知识有效地整合到大语言模型中，如何有效地缩减模型请求的次数和大小，同时保持任务执行的准确性和完整性。可将问题抽像为以下进一步设计和实施几个方面：

• 界面表示(GUIRepresentation)：在这个项目中，系统需要处理的输入输出均为应用的图形界面的状态和动作，而并非大语言模型通常擅长处理的自然语言表达。因此为了充分利用大预言模型在自然语言上的理解能力和推理能力，应用的图形界面、状态以及动作，都需要通过合理的方式转换为自然语言描述的形式，同时确保信息传递的高效性。

• 知识整合(KnowledgeIntegration)：为了提升大语言模型的表现能力，我们需要设法提供和当前操作的应用相关的信息，这些信息将在LLM决策时发挥参考作用。

• 开销优化(CostOptimization)：使用大语言模型进行推理所花费的开销是巨大的，同时也会耗费相当的计算资源；与此同时，使用LLM实现任务自动化可能带来超长的提示词。因此，如何优化对LLM的请求效率(QueryEfficiency)是很有必要的。

综上，将1.1中智能手机助手中4个方面的问题抽象为能够通过计算机技术解决三个方面的功能规划。

## 问题定位

技术领域：该项目人工智能、自然语言处理、大模型。

## 问题评估

（1）首先，大语言模型具有推进任务自动化这一领域继续发展的巨大潜能。目前，声控智能个人助手(IPA, Intelligent Personal Assistant) 是移动端设备应用自动化的典型应用场景，旨在为用户提供智能、高效、自动化的使用体验。IPA不仅仅可以部署在手机端，在车载信息娱乐系统(IVI,In-VehicleInfotainment )、可穿戴健康监测设备(WearableFitnessTrackers ) 以及各类AR/VR设备上。一般情况下，开发者需要手动配置整个IPA服务的工作流，而这很可能是十分冗杂的。近来也有研究者试图构建可以像人类一样直接操作GUI界面元素的系统；但是这些系统往往需要人工描述，或者是提供步骤清晰的任务。如果我们能够利用LLM的语义理解和推断能力，应该对系统在GUI任务上的表现有所裨益。

（2）其次，将大语言模型嵌入到智能设备也会极大提高大语言模型本身的表现。大语言模型的训练数据来自于大规模的常识性的公共数据，但是它对单独的某个用户的个人信息和使用习惯几乎一无所知。而智能手机则是个人信息的一个良好载体，承担着每个人日常进行的各项活动。如果大语言模型能够学会挖掘这些个人信息并在完成任务时加以利用，这将会极大提高它作为个人智能助理的能力和优势。

## 问题分解

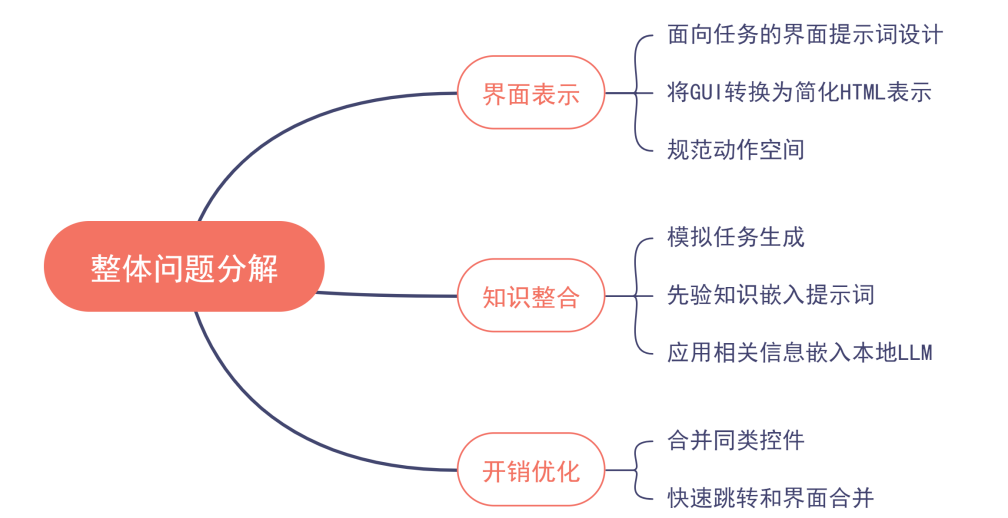


图 1 问题分解

依据1.2中的问题抽象，项目主要问题为界面表示、知识整合、开销优化。对问题进一步划分如图1所示。整体分为三个大模块，分别描述如下：

• 面向任务的界面提示词设计：界面提示词设计(UIPrompting)指的是用文字描述界面信息，并将该文字描述嵌入到对大模型请求 的提示词中。界面提示词设计的目的是为大模型清楚明白地展示当前界面的一些文字性的或者结构性的内容，同时限制大模型的输出格式，以关注特定的反馈结果。接下来我们用图2表示整个界面提示词设计的过程。

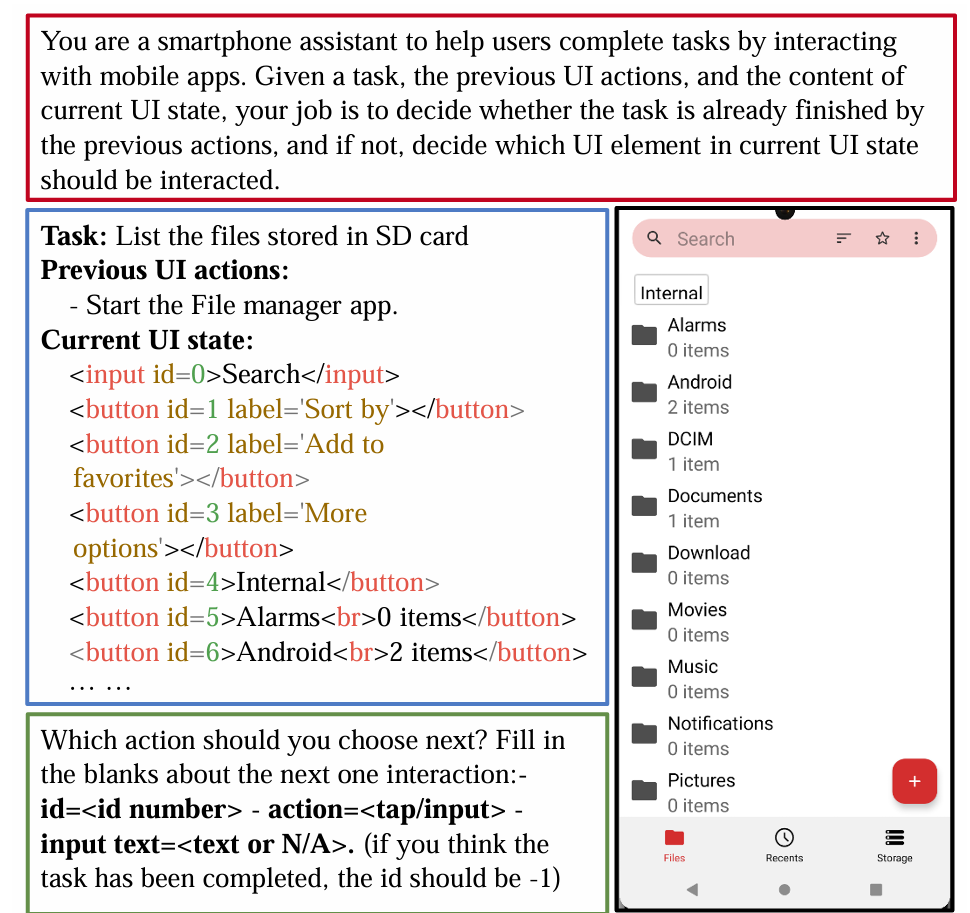


图 2 UI Prompting 具体过程

• 基于探索的知识整合：基于探索的知识整合(Exploration-based Memory Injection)的目的在于为大语言模型提供和当前应用直接相关的信息，让大语言模型能够利用这些信息，理解当前应用的运行逻辑，得出更为有效的决策。在这一方面我们仍然注意到了以下几点困难：（1）随机探索得到的界面转换关系图(UTG,UITransitionGraph)无法直接交由大模型进行处理；（2）直接从界面跳转的自动化agent只能获取到随机探索过程中每一步的UI界面和采取的动作，和实际完成某个特定任务有很大区别；（3）应用所具有的界面数和控件数非常多，如果一次性交给大模型，将超过大语言模型的token数限制。为了解决上述问题，AutoDroid从随机生成的UTG中无监督生成任务；而这些生成的任务将作为参 考，指导大语言模型如何完成真实的用户任务。

• 多粒度请求优化：在AutoDroid 运行时，主要的开销来自于对大语言模型的请求。因此，降低每个任务的请求频次将有助于降低整个系统的开销。此外，在更细粒度的层面上，在每个提示词中删去不必要的部分，也会是相当有效的方法。

# 相关工作

1. 用户界面（UI）任务自动化：

用户界面任务自动化是指建立能根据用户界面自动执行任务的系统。最相关的工作可分为以下几类：1.理解图形用户界面的语义。这些方法可以帮助总结图形用户界面的主要功能，理解每个控件的目的，并解决与图形用户界面相关的用户问题。这对实现用户界面自动化和无障碍等各种交互任务非常重要。2.将用户指令映射到用户界面元素。这些方法旨在选择与用户指令最相关的图形用户界面元素。最近，一些研究正在使用 LLM 将指令映射到图形用户界面元素。3. 移动用户界面任务自动化。这些方法建立了一个代理，通过在图形用户界面上执行操作来完成用户的任务。另一方面，AutoDroid利用自动生成的用户界面过渡记忆来完成高级任务，包括智能手机上的多个步骤。该记忆可以帮助代理理解信息丰富的用户界面和应用程序的使用，并在推理和规划中增强 LLM。

1. 增强型 LLM：

虽然 LLM 可以执行问题解答和文本生成等各种任务，但它们仍然受到固定权重集和最大上下文长度所能存储信息的限制。因此，研究人员正在尝试用各种工具来增强 LLM，如网络浏览器、各种 API，甚至其他模型。这些研究通常依靠巧妙设计的提示来提高 LLM 的推理能力，或在特定领域的数据集上对 LLM 进行微调。此外，它们还经常假设有公共可用的应用程序接口来解决任务。而我们的方法并不依赖人工设计的应用程序接口，因为这些应用程序接口在移动应用程序中并不常见。

1. 自动应用程序分析：

自动应用程序分析是使用软件工具自动分析 Android 应用程序的功能、性能和可用性的过程。典型的技术包括静态分析（分析移动应用程序的静态组件，如代码、资源文件和配置）和动态分析（侧重于运行时行为）。在移动任务自动化中，关键是要了解应用程序的功能。先前的工作已经探索了基于代码、图形用户界面、网络流量和元数据来提取此类功能信息，但它们大多过于粗粒度，无法促进任务的完成。我们的方法使用 LLM 处理动态应用程序分析器收集的原始痕迹，以获得每个图形用户界面元素的细粒度描述。

# 技术方案

## 技术方向

大模型+AIOT

## 技术选择

大模型：我们利用GPT-4生成模拟任务存储在本地，执行任务时个人助手可以先查询本地记忆，从而加快任务执行。考虑到访问云端大模型的网络延迟以及本地隐私的安全性，我们在本地运用MLC技术部署了Vicuna-7B，并对其进行了微调，使其更好的适配任务的执行。Vicuna是开源大模型LLama的变种，正确率高于LLama。

模拟点击与输入：Android Accessibility（安卓无障碍服务）旨在帮助用户改善设备的可访问性，使得那些有视觉、听觉、运动或认知障碍的用户也能更容易地使用设备。无障碍服务可以在后台监控用户界面的事件，进行模拟点击、输入等功能。

隐私数据检测与替换：通过模式识别、实体检测、模式验证、基于上下文提高检测信度等步骤，快速识别并替换文本中的敏感数据，如：信用卡号码、姓名、位置、身份证号码、电话号码等。将用户隐私信息用非专用词取代（例如“＜姓名＞”→“张三”），在收到LLM的响应后，再将特殊单词映射回原始单词。从而保护用户的隐私信息。

大模型本地部署：MLC技术。MLC是一种高性能的通用部署解决方案，在其中已经内置了量化技术，可以在尽量少的降低大模型正确率的情况下大幅减少大模型参数，部署本地大模型量化是非常关键的一步，因为手机的内存消耗是非常重要的。MLC还会对模型进行部署优化，使其的代码和内存管理等方面更加适合手机这个载体。

## 结果期望

（1）实现任务自动化功能。用户可以通过语音输入想要完成的任务，app会自动生成任务完成的步骤并执行，将任务完成。遇到危险任务app会进行提问，从而避免对用户的权益造成损害。

（2）实现隐私保护功能。基于Microsoft Presidio算法，在线处理过程中，AutoDroid运行一个人身份信息（PII）扫描仪，检测UI信息中的用户隐私信息，包括姓名、电话号码、电子邮件地址等。将用户隐私信息用非专用词取代（例如“＜姓名＞”→“张三”），然后将Prompt发送到云。在收到LLM的响应后，AutoDroid在解析操作之前将特殊单词映射回原始单词。

（3）实现无障碍服务功能。应用通过模拟人类的点击和输入等操作，直接与手机应用的图形用户界面（GUI）互动。这种方法不仅提高了安全性和隐私性，还确保了agent能够适应应用界面的变化和更新。

（4）实现大模型本地部署。基于大语言模型的高性能部署解决方案MLC LLM，将具有强大性能的开源大语言模型Vicuna部署到手机上，从而调用本机上的LLM来进行移动任务动作的推理，减少query大模型所需的时间。

# 技术实践

## 使用的开发框架及依赖的库

项目采用深度学习+大模型+物联网+安卓框架的实现方案，以深度学习为基础，制作合适的数据集对Vicuna进行微调，运用合适的算法对自动化过程进行精简并形成记忆，采取APP的服务提供方式与用户进行交互。用户可通过App对手机实现自动化控制，同时建立隐私保护机制，保护用户的隐私。

主要程序代码：

* tansorflow：提供了一个灵活的平台，使得开发者可以轻松构建、训练和测试各种机器学习模型，包括但不限于传统的线性回归、逻辑回归模型，以及复杂的深度学习模型，如卷积神经网络（CNNs）、循环神经网络（RNNs）、长短期记忆网络（LSTMs）、生成对抗网络（GANs）等。
* Gym：OpenAI 开发的一个流行的强化学习工具包。它主要用于创建和研究强化学习算法，为开发者提供了一个标准化的环境接口，使得算法开发者能够在一个统一的架构下测试和比较不同的强化学习算法性能。
* Imageio：用于读写多种图像和视频格式的 Python 库。
* Telnetlib：用于实现 Telnet 协议的客户端功能。
* Socket：允许开发者直接使用 TCP/IP 协议族进行网络编程。
* Subprocess：用途是管理和控制子进程的创建、执行以及与其交互。
* Logging：用于提供详细的日志记录功能。
* Traceback：用于处理程序运行时出现的异常和错误信息。
* time：用于处理时间。
* json: 用于处理JSON数据。

APP端：

* AccessibilityService：用于创建辅助功能应用，它能够监听并响应设备上的用户界面交互事件，帮助残障人士或有特殊需求的用户更方便地操作和获取应用程序的信息。
* Wedget：Android SDK中提供的一组核心UI组件集合，它是构建Android应用程序用户界面的基础。这个库包含了多种预定义的视图（View）和视图组（ViewGroup），这些控件可以用于显示文本、图像、列表、按钮等元素，并允许用户与之交互。
* Content：提供了应用程序之间进行数据交换和通信的基础机制。这个库包含了关键类和接口，用于处理Intent、内容提供者(ContentProvider)、广播接收器(BroadcastReceiver)以及应用上下文(Context)等功能。
* View：定义了所有图形和事件处理的基础组件。这个库提供了与屏幕显示、用户交互以及窗口管理相关的重要类。
* Os：为应用程序开发者提供了与底层操作系统的交互能力。这个库中的类和接口主要用于处理进程间通信（IPC）、线程管理、消息传递、系统服务访问以及一些高级的系统功能。
* Java.utils：提供了大量的工具类和数据结构，旨在增强Java应用程序的功能性和实用性。
* Okhttp3：用于在Android和Java应用中执行网络请求。
* Gson：用于在Java对象和JSON数据之间进行序列化和反序列化操作。

## 技术实践过程

1. 主要程序

* 随机探索模块：

在离线阶段AutoDroid会运用广度优先遍历算法对手机中的APP进行探索，生成相对应的UI Transition Graph，并对UI组件进行剪枝操作，对于不同界面中出现的重复的组件，仅保留离初始界面最近的一个。完成此操作会使我们的APP对将要自动化操作的APP有一个大致的了解。然后经过剪枝之后的所有组件通过query LLM的方式获取其功能，我们称之为模拟任务。

* 记忆生成模块：

通过将UI界面转换为HTML语言描述格式，可以将页面和探索中的点击输入给ChatGPT，由ChatGPT根据输入内容生成虚拟任务作为APP Memory，所有的APP Memory中的模拟任务会通过一个特征提取模型Instructor-XL映射成固定维度的特征向量。在用户输入指令以及执行任务的APP后，系统首先会通过Instructor-XL将用户输入指令做映射，然后和该APP Memory中的模拟任务计算相似度。挑选相似度最高的模拟任务对应的组件，在任务执行开始的时候，先行跳转到该组件对应的界面并点击该组件。这样的设计减少了执行指令需要query LLM的次数，减少了开销，也减少了完成指令的所花费的时间。

* 隐私过滤器：

APP基于Microsoft Presidio算法，在在线处理过程，AutoDroid运行一个人身份信息（PII）扫描仪，检测UI信息中的用户隐私信息，包括姓名、电话号码、电子邮件地址等。将用户隐私信息用非专用词取代（例如“＜姓名＞”→“张三”），然后将Prompt发送到云。在收到LLM的响应后，AutoDroid在解析操作之前将特殊单词映射回原始单词。

* 微调后本地部署的Vicuna：

将任务直接询问ChatGPT是一种相对简单的处理方法，但是相应的会增加时间成本。为了使手机个人助手运行速度加快，我们需要运用MLC将大模型部署在本地，以去除网络延迟的影响。MLC是一种成熟的高性能的通用部署解决方案，它可以通过对大模型的代码和内存管理等方面进行优化，使其更加匹配手机硬件的运行，加快模型的推理速度。并且它还会对模型进行量化处理，在尽量不降低正确率的情况下大幅减少模型的参数量，从而减少大模型占据的内存，有利于在手机这种内存较少的边缘设备上运行。

1. APP：

* 应用读取：

在应用启动时，首先从系统中获取所有可作为主入口点的应用程序信息，包括其显示名称、包名和启动类名，并按照名称排序。这些信息被存储在集合中，并用于填充Spinner控件作为列表显示，让用户能够选择一个已安装的应用。

* 无障碍服务检查：

检查应用自身的无障碍服务（MyAccessibilityService）是否已经启用。如果已启用，则直接启动该服务；否则，引导用户进入系统设置界面开启无障碍服务。

* 初始化与配置：

无障碍服务在启动时首先进行必要的初始化，包括读取配置信息，确定需要关注的应用程序包名列表。根据配置信息注册并设置服务监听事件类型及范围，确保能够捕获指定应用内的用户界面交互事件。

* 广播接收与响应：

内置一个广播接收器来监听自定义的广播消息，一旦接收到特定的任务指令，服务将触发相应的处理逻辑。处理逻辑可能包括解析任务详情、准备执行动作等环节。

* UI事件监听与分析：

在系统层面持续监控关注应用的用户界面事件，如按钮点击、文本输入等，并实时抓取当前活动窗口中的UI元素结构及其属性。

* 智能决策与交互模拟：

将获取到的UI信息与任务需求相结合，通过内置或外部AI管理模块（如OpenAIManager）进行分析和决策，识别出下一步操作的目标UI元素。使用无障碍服务功能模拟用户的实际操作，例如模拟点击、滑动或者输入文本等行为，以自动化完成预设任务。

* 状态检测与循环执行：

在执行操作后，服务会根据预先设定的规则判断任务是否已经成功完成。若任务未达成，服务将继续循环上述过程，不断收集新的UI信息并重新决策，直到满足任务完成条件为止。

# 结果验证

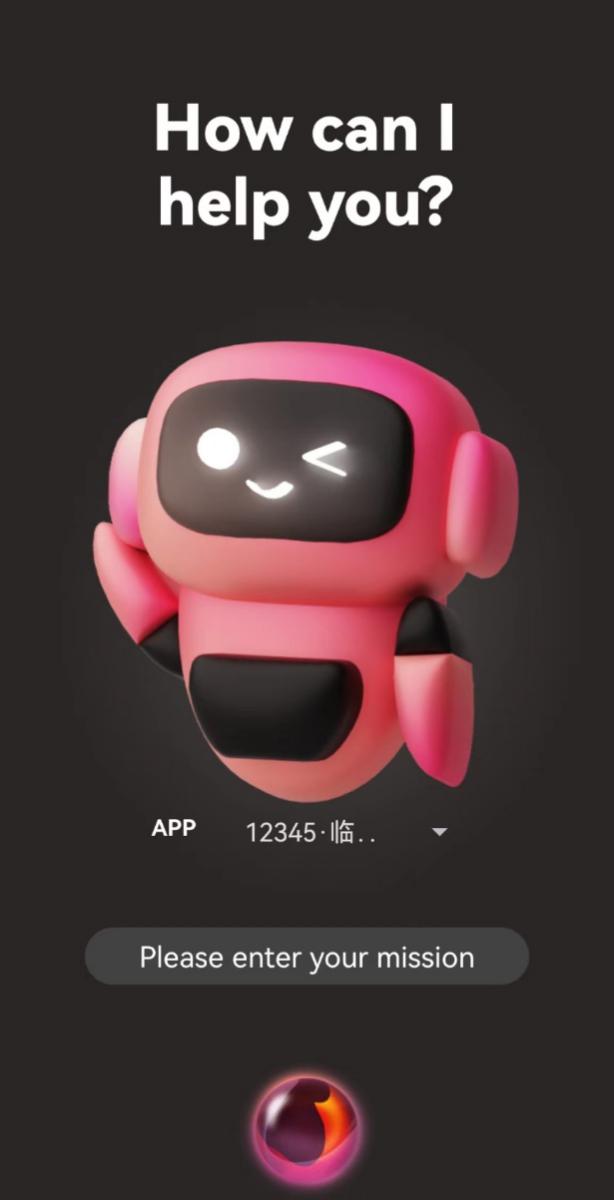


图3：AutoDroid 启动界面

AutoDroid前端展示，会有三个主要的组件：APP选择的下拉框，任务输入框以及开始任务按钮。



图4：APP选择界面

AutoDroid APP选择下拉框，会展示当前APP上的所有APP名称。