（题目，缺）

（摘要，缺）

1. 介绍

该部分暂缺，因主体立意和技术创新点未确定

1. 背景

2.1 移动端任务自动化的动机及含义

2.1.1 移动端任务自动化的动机

（此处欠缺一个更明确的动机。思考后认为以下几个方向合适：辅助老年人或残障人士使用应用、特殊应用场景（例如健康码app不统一增加的学习成本）、优化交互体验，留待讨论）

2.1.2移动端任务自动化的含义

**移动端任务自动化**，是接受用户以自然语言描述的任意任务以及一个用于执行该任务的移动应用程序，并以输出一些列可以执行的用户界面动作的形式自动完成用户交代的各种任务。

**任务**是用户提出的功能性请求，旨在在移动设备上针对某一个移动应用程序执行一系列操作。请求通常涉及多个步骤，且可能没有明确的指示。为使智能手机系统能够自动化任务，需要进行语义理解和推理，以将任务拆解为个别的可执行命令。

**UI状态**指的是用户在使用移动设备时看到的屏幕状态。它由一系列由图像和文本组成的控件组成，通常表示为GUI树。为了执行任务，用户或其它代替交互的程序需要提取并理解当前UI状态中的信息。

**UI动作**是用户可以在移动设备上执行的动作，表示为元组（目标元素，动作类型，值）。目标元素指的是UI状态中的一个控件，如按钮、文本框、输入字段或滑块。动作类型表示对目标元素的操作方式。为简便起见，我们考虑三种主要的智能手机交互类型，包括“点击”、“输入”和“滑动”。值字段指的是“输入”动作的文本内容，对于其他动作类型则为空。

大多数针对移动端任务自动化的现有方法需要开发者/用户做出大量努力才能实现自动化任务。而我们旨在实现无监督任务自动化，即在不需要手动调整的情况下支持在黑盒应用程序（即内部机制未知的应用）上自动化任意任务。不过，我们假设这些应用程序可供自动化分析，例如探索状态、爬取内容和分析代码。这样的假设是合理的，因为应用程序包都可以下载，且静态/动态应用程序分析技术在此前已经被广泛研究[10, 30-33]。

2.2 大语言模型

大型语言模型（简称LLM）主要指基于Transformer[57]架构，包含数十亿参数并在大量文本数据上训练的语言模型，例如ChatGPT[42]、GPT-4[43]、PaLM[7]、LLaMA[55]等。这些模型展现出小型模型所不具备的能力，包括逻辑处理[8]、程序生成[5]和多步推理[61, 65]。一般来说，LLM能够比在专用数据集上训练的基准模型更好地执行任务。

目前，研究人员正付出诸多努力，试图将LLM和工具结合在一起，例如调用API等。然而，在以移动端任务自动化为目标时，这些努力并不能做简单的移植。这是因为移动端任务自动化的情况更加复杂，而且海量的、不同的APP缺乏公开API文档。因此，我们尝试寻找另一种方法。然而，在寻找别的方法的过程中面临着几个挑战，包括GUI表示、知识整合和成本优化。

**GUI表示**：LLM仅能处理纯文本数据，不能直接处理GUI状态，也不能直接返回UI动作的输出。尽管Android中的GUI状态可以使用UI层级查看器等工具表示为文本，但通常这种表示非常冗长（平均每个UI状态约40k tokens），很容易超出LLM的处理范围。

**知识整合**，LLM缺乏对某些应用的知识和经验，这可能导致指令执行不正确。

（此处缺一个图及一系列说明，大概内容是仅靠当前UI状态的上下文无法确定任务的具体含义，必须结合其他层次的UI状态才能确定。例如，B站中，点击关注一定得发生在点开视频之后）

解决这个问题的一个可能方法是先探索和分析移动应用，获取经验，然后尝试为用户完成任务。

**成本优化**，使用LLM完成任务可能导致极高的成本。根据[44]，查询ChatGPT API[42]的价格为1.5美元/1000K tokens。即便我们可以部署一个私有LLM服务，计算成本仍然高昂。例如，使用LLaMA-7B[55]推断单个tokens需要6.7亿FLOPs。

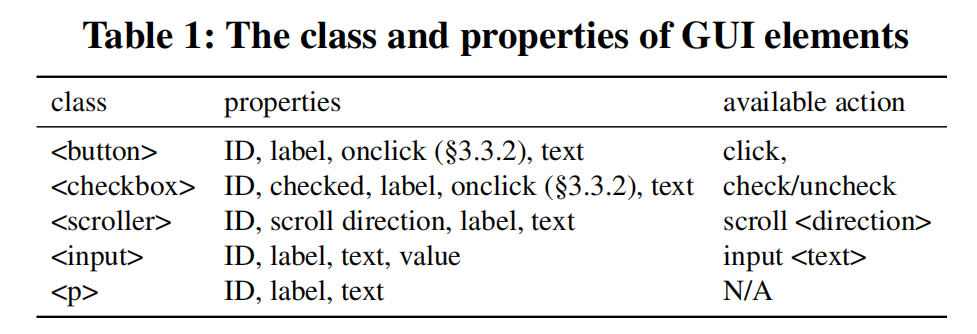
3. DroidBoot实现方法

3.1 任务驱动的UI表示方法及其与LLM的结合

我们使用新的UI表示方法，旨在将底层UI信息以文本形式表示作为LLM的输入，向LLM清晰地展示UI的文本和结构内容，并限制LLM的输出，仅预测有效的UI交互。

3.1.1 将GUI转换为简化的HTML表示。我们使用了一个GUI解析模块，将GUI转换为LLM能够处理的简化HTML表示形式。[58]的作者发现，由于LLM的训练数据中含有大量的HTML代码，LLM在理解HTML表示的UI界面时表现得比理解使用自然语言描述的UI界面要好。因此我们选择以HTML风格呈现GUI。

一共有五种类型的HTML标签，分别是<button>、<checkbox>、<scroller>、<input>和<p>，分别代表可以被点击、勾选、滑动、编辑和表示其他视图的元素。每个元素包含的属性有：ID（元素在GUI树中出现的顺序）、label（描述元素功能的内容描述）、onclick（关于点击此按钮或勾选/取消勾选此复选框后将访问的UI状态的提示）、text（元素上的文本）、direction（滚动方向，包括上/下/左/右）、checked（复选框是否被勾选）、value（已输入到文本框中的文本）。GUI元素的类别和属性在表1中展示。



3.1.2 受限的操作空间

UI任务自动化的一个关键特征是，所有UI动作必须受到底层应用的约束，即用户或代理程序只能在现有的UI元素上执行支持的操作类型。因此，我们需要针对LLM的生成性质做出一定改变，将其应用于这种离散选择任务。一种方法是强制LLM产生的结果必须符合预定的结构，例如要求LLM的结果必须符合以下结构：- id=<id> - action=<tap/input> text=<text or N/A>（任务完成时，id=-1）。

3.2 记忆化的自动探索

记忆化的自动探索旨在向大语言模型提供与应用的背景信息，使它们能够获取应用程序的相关知识，理解应用程序的使用方法，并作出有效的决策。然而，简单输送背景信息以协助LLM进行任务自动化存在诸多挑战，包括：

1. 通过随机探索获得的用户界面转换关系图（UTG）不能被LLM直接处理。
2. 仅在用户界面进行自动化探索，获得的知识只包含用户界面和动作数据，缺乏直接实现任务自动化所需的关键信息。例如完成特定任务所必需的特定用户界面元素和动作的细节。

（iii）一个应用可能有众多的用户界面屏幕和用户界面元素（按钮、文本框等），如果将它们全部包含在一个提示中，可能会超过LLM的tokens长度限制。为了克服这些挑战，AutoDroid基于随机探索的用户界面生成模拟任务。这些模拟任务将会成为指导LLM如何完成用户任务的指南。

3.2.1 模拟任务的生成

AutoDroid通过分析用户界面转换关系图（UTG）来生成模拟任务。用户界面自动化探索工具生成的UTG包含了有关应用程序的重要信息，如用户界面之间的连接和每个屏幕上不同用户界面元素的存在。通过总结所有用户界面元素的功能，我们可以全面了解应用程序内可以执行的任务，并确定执行它们所需的相应用户界面元素。因此，AutoDroid解析UTG中所有用户界面状态和用户界面元素，并通过查询LLM提取它们的功能。

具体来说，UTG可以被视为一个有向图，其中节点和边分别是随机探索者记录的所有用户界面状态和动作，分别用U和A表示。对于每个用户界面状态U\_i，利用LLM以总结所有用户界面元素{e\_ij}|U\_i|j=1的功能（其中|U\_i|表示e\_ij中元素的数量）。需要注意的是，如果某个元素在多个用户界面状态上出现，AutoDroid只根据最靠近初始用户界面状态的结点提取该元素的功能。遍历UTG中所有用户界面元素后，我们获得一个包含n个条目的模拟任务列表（其中n代表UTG上用户界面元素的总数）。表中的每个条目对应一个用户界面元素e\_ij，并分为三个部分：<模拟任务总结，用户界面状态，用户界面元素>。“模拟任务总结”代表由LLM总结的e\_ij的功能，代表点击此元素可完成的模拟任务。“用户界面元素”包括从应用的初始用户界面开始点击的所有元素，直到达到U\_i。“用户界面状态”代表从初始用户界面状态到U\_i遍历的用户界面状态序列。这个列表为后续的代理交互程序提供了实现每个功能所需操作的信息，帮助它们规划如何有效完成给定任务。除了模拟任务表外，还有一个额外的表格称为用户界面功能表。它总结了UTG中每个用户界面状态的功能。

3.3 思维链或可能的优化，缺

4. 评价，缺，数据部分亦放在此处。

5. 相关工作

（该部分我觉得也可以借鉴那两篇论文）

6. 结论