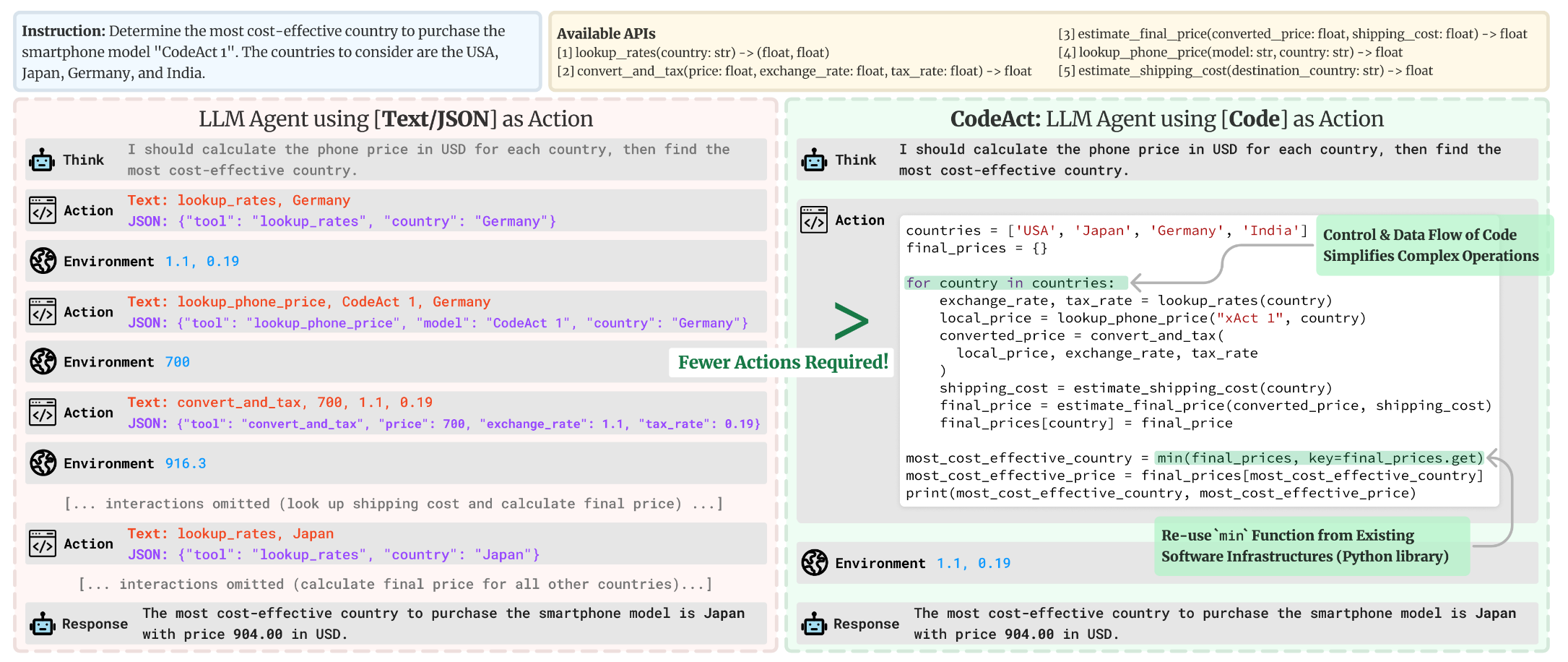
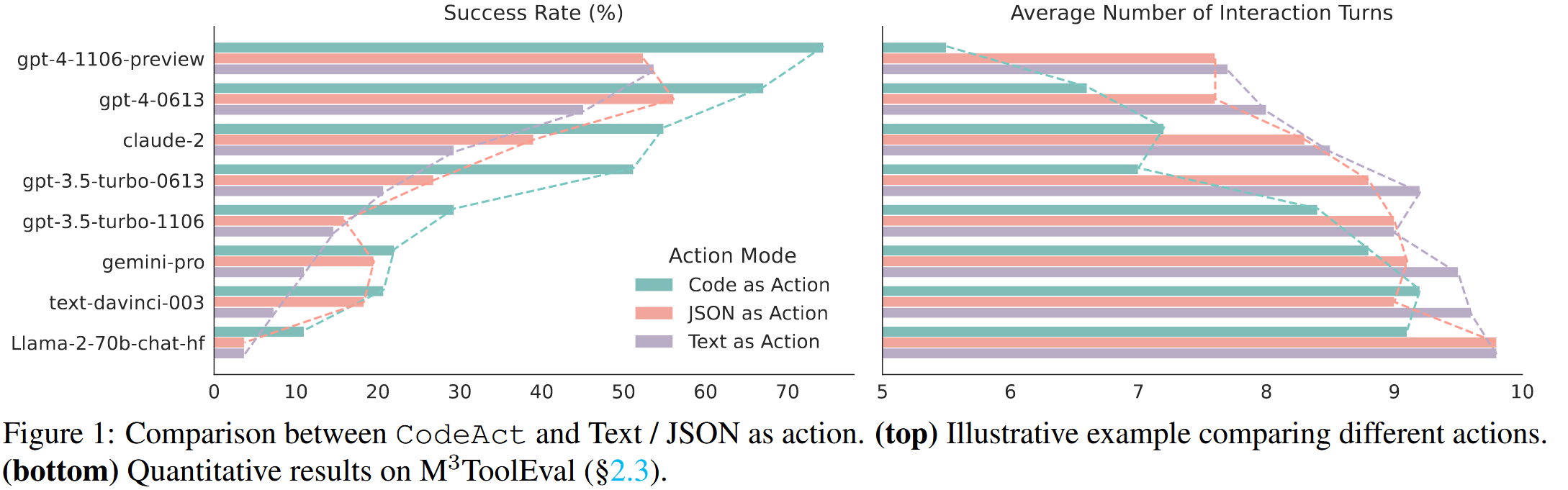
# 论文分享：Executable Code Actions Elicit Better LLM Agents

## 摘要

基于大语言模型（LLM）的Agent能够执行广泛的操作，例如调用工具和控制机器人，在解决现实世界的挑战方面显示出巨大的潜力。**LLM Agent通常通过以预定义格式生成JSON或文本来提示生成操作**，而这种格式通常受到**受限的操作空间**（例如，预定义工具的范围）和**受限的灵活性**（例如，无法组合多个工具）的限制。本工作建议**使用可执行的Python代码将LLM Agent的动作整合到一个统一的动作空间（CodeAct）中**。与Python解释器集成后，CodeAct可以执行代码动作并动态修改先前的动作，或者通过多轮交互在新观察到的情况下发出新动作。我们对APIBank上的17个llm的广泛分析和新策划的基准表明，CodeAct优于广泛使用的替代方案（成功率高出20%）。





## CodeAct的优势

1. 与Python解释器集成，CodeAct可以执行代码动作并动态调整先前的动作或根据通过多次交互接收到的Observation（例如，代码执行结果）发起新动作。
2. 代码操作允许LLM利用现有的软件包。CodeAct可以使用现成的Python包来扩展操作空间，而不是限于人工定制的特定于任务的工具。它还允许LLM使用大多数软件中实现的自动反馈（例如，错误消息），通过自调试其生成的代码来改进任务解决。
3. 代码数据被广泛用于预训练当今的LLM。这些模型已经熟悉结构化编程语言，因此可以有效地采用CodeAct。
4. 与JSON和具有预定义格式的文本相比，代码本质上支持控制和数据流，允许将中间结果存储为可重用的变量，并允许使用一段代码来组合多种工具，以执行复杂的逻辑操作（例如，if语句，for循环），从而释放llm通过利用其编程知识来解决复杂任务的潜力。在图1中，使用CodeAct（右上）的LLM可以将相同的工具序列（例如，将一个工具的输出作为输入传递给使用数据流特征的另一个工具）应用于通过for循环（即控制流特征）的所有输入。而文本或JSON必须对每个输入执行操作（左上角）。

总结：与多数现有的LLM智能体不同的是，CodeAct的突出之处在于：能够充分利用现有LLM对代码数据的预训练，以实现低成本高效的采用。而且本质上可以通过控制和数据流支持复杂的操作，还可以使用广泛的软件包来扩展行动空间和自动反馈。

## CodeActAgent框架

