

# 大语言模型引导策略 实验与分析

第七组

李鹏达 10225101460

武泽恺 10225101429

张耘彪 10225101437

## 1 简介

在对大语言模型引导策略进行进一步修改后，我们再一次进行了初步的实验——对随机策略 (random)、原有大语言模型引导策略 (llm) 和我们提出的大语言模型引导策略 (new) 三种策略在两个开源应用程序 Ominotes 和 AnkiDroid 上进行测试。我们使用覆盖率作为评估指标，并将测试结果与随即策略和原有大语言模型引导策略进行对比，以验证我们提出的新策略的有效性。

## 2 实验环境

- 操作系统：Windows 11 家庭中文版 26100.3915
- Python 版本：3.11.9
- JaCoco 版本：0.8.10
- Android Emulator 版本：Pixel 9 API 34

## 3 实验方法

在实验中，我们使用 JaCoco 对安卓应用程序进行插桩，并使用 Android Emulator 进行测试。使用每个策略运行 Kea 至少 120 分钟或 1200 个事件，在测试过程中，每两分钟记录一次覆盖率数据。

实验结束后，分别根据运行时间和时间次数两个维度对覆盖率数据进行分析。我们将每个策略的覆盖率数据进行对比，并计算出每个策略的平均覆盖率和标准差。

我们的实验脚本可以在 <https://github.com/llipengda/kea-test> 中找到。

我们插桩的应用程序可以在 <https://drive.google.com/file/d/1XM7E4WIdaogFF00z8kTrq3MsLTD-7p/view?usp=sharing> 下载。

## 4 实验结果

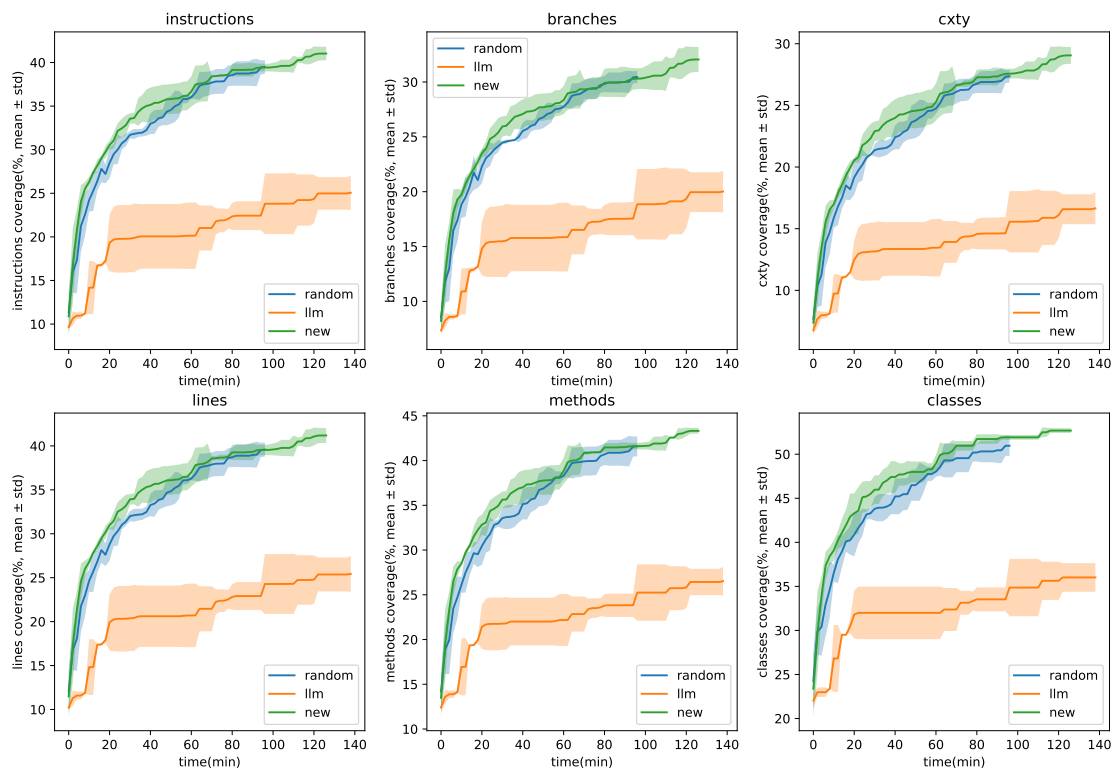


图 1: OmniNotes 覆盖率数据, 按时间

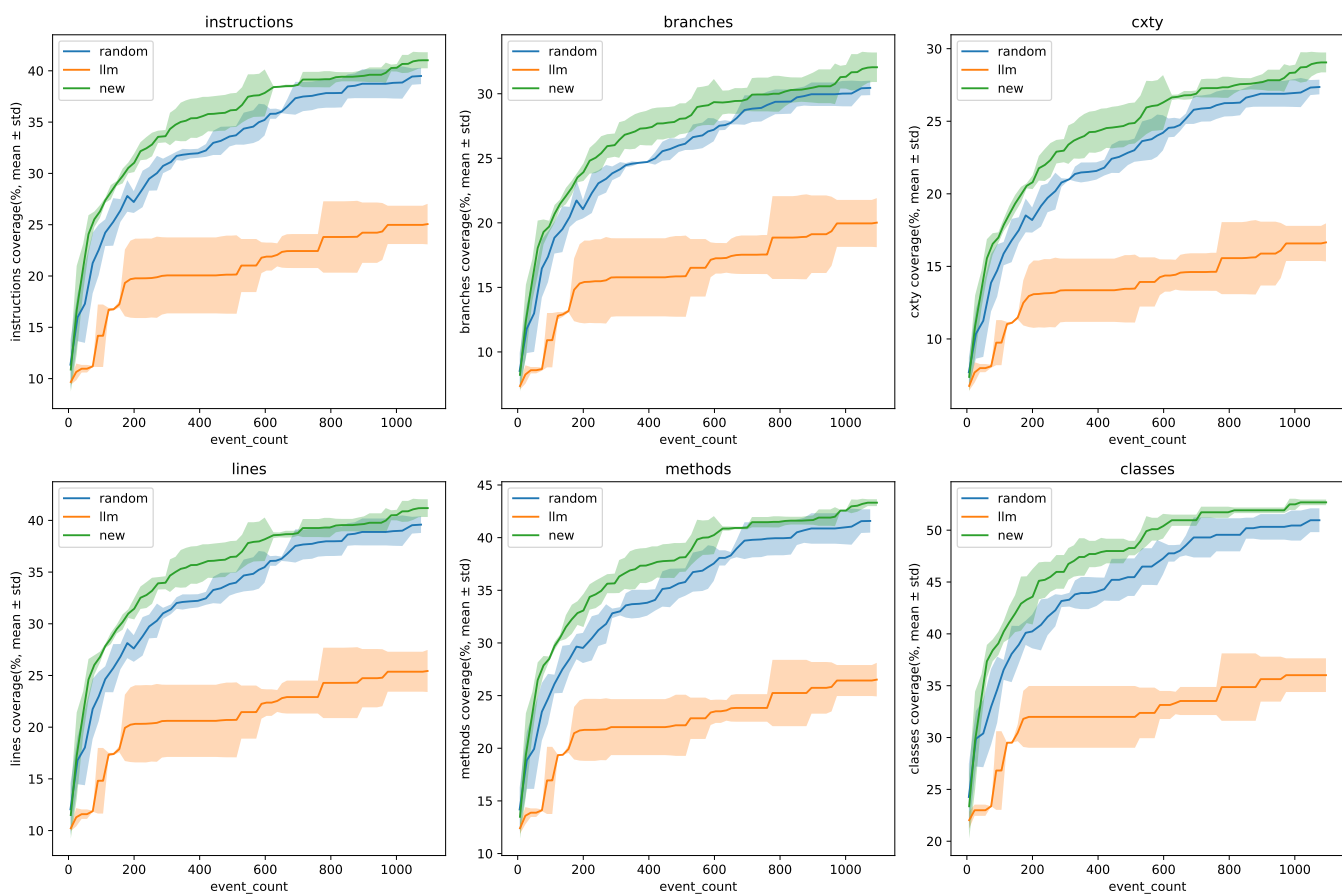


图 2: OmniNotes 覆盖率数据, 按事件数

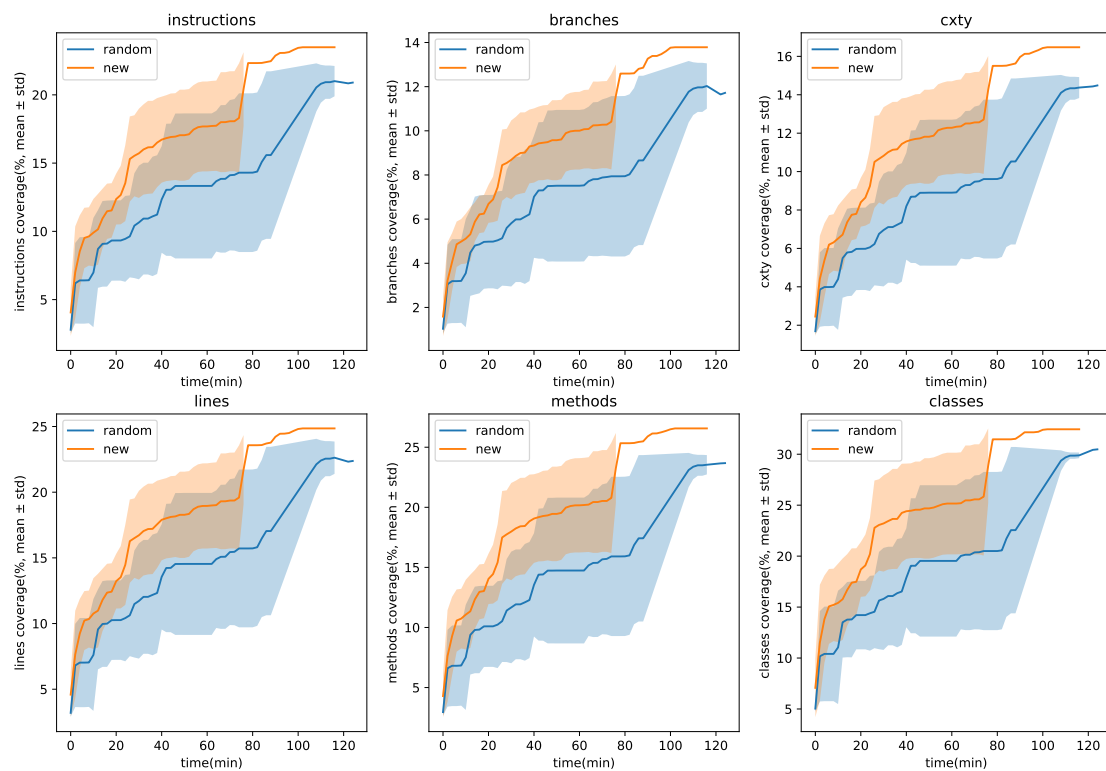


图 3: AnkiDroid 覆盖率数据, 按时间

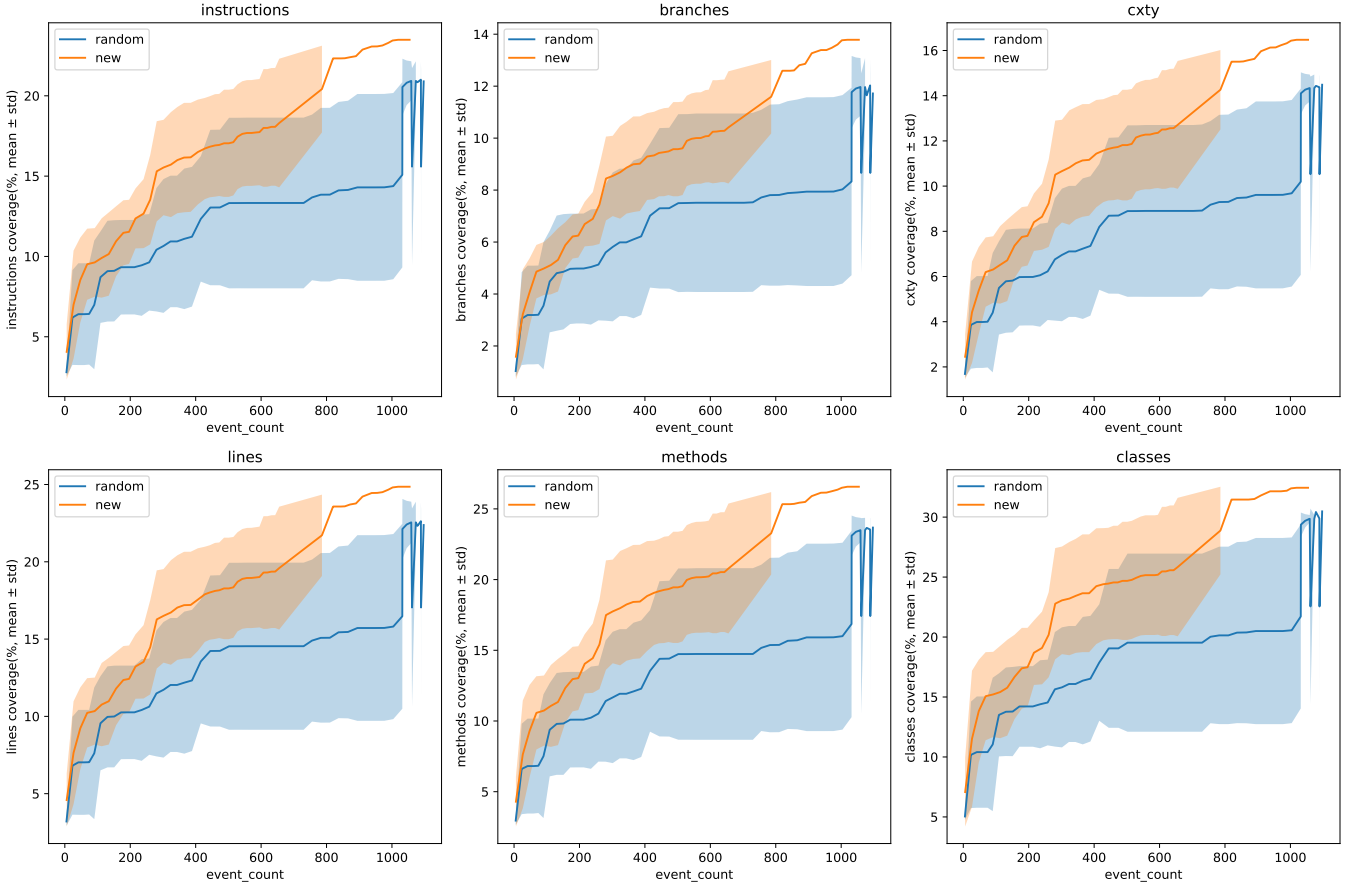


图 4: AnkiDroid 覆盖率数据, 按事件数

## 5 实验分析

实验数据表明, 对于 Ominotes 应用程序, 我们的策略覆盖率在时间维度上较随机策略略有提升, 大幅优于原有大语言模型引导策略; 在事件数维度上, 我们的策略覆盖率较随机策略有一定提升, 且提升大于时间维度, 大幅优于原有大语言模型引导策略。这说明我们的策略在运行相同事件时能够得到更高的覆盖度。但由于引入了大语言模型, 导致运行速度有所下降, 因此在时间维度提升不明显。查。

而对于 AnkiDroid 应用程序, 无论是在时间维度还是事件次数维度, 我们的策略均对于随机策略有明显的提升。经过对实验数据的分析, 我们猜测这可能是由于该应用程序中含有 WebView 和登录页面, 导致随机策略更容易陷入 UI 陷阱。

值得注意的是, 无论是哪个应用程序, 哪种策略, 得到的覆盖度都是比较低的, 而且观察可以发现它们均有大量的页面没有探索。