# GEM 复现及后续工作的说明文档

Yang

2025 年 7 月 24 日

## 1 实验介绍

GEM 是区分数据集中 ID 和 OOD 的工具，本次实验是在复现 GEM 的基础上进一步探究模型在 ID 和 OOD 数据集上的表现，以此探究模型在分别进行 SFT 和 RFT 之后对性能的影响。

实验任务：

- 对测试集使用 GEM 进行划分得到带有标签”label” 的数据集
- 使用基础模型、SFT 模型和 RFT 模型分别进行单步推理及结果评估，最终结果中会在原测试集上”type match” 和”exact match” 两个真值标签
- 上述两步结果都是在测试集上添加新的关键词，统计 ID 情况下”type match” 和”exact match” 的平均值以及 OOD 情况下的，进行对比，探究 SFT 和 RFT 对性能的影响

实验具体操作：

- 首先运行 run.py 分别得到 train inputscore 和 test inputscore 两个文件
- 然后运行 GEM.py 得到经过 ID 和 OOD 分类的测试集 dataset.json
- 运行 run predict.py 得到不同模型在同一测试集下的表现结果，保存为 result.json 和 summary.json
- 运行 count script.py 统计任务三的结果

实验变量：

- 模型种类：
  - Qwen2-7B-Instruct(base)
  - UI-TARS-7B-SFT(SFT)
  - UI-TARS-7B-DPO(RFT)
- 用作 GEM 划分的模型：Qwen 和 UI-TARS

## 2 实验结果

统计后比较 Qwen 和 UI-TARS 在 ID 和 OOD 上划分的差异并统计为 Venn 图：

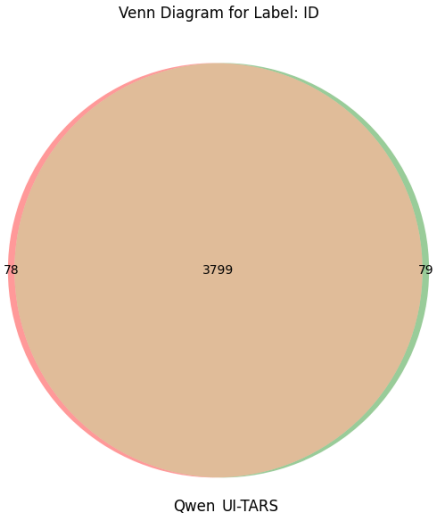



图 1: ID

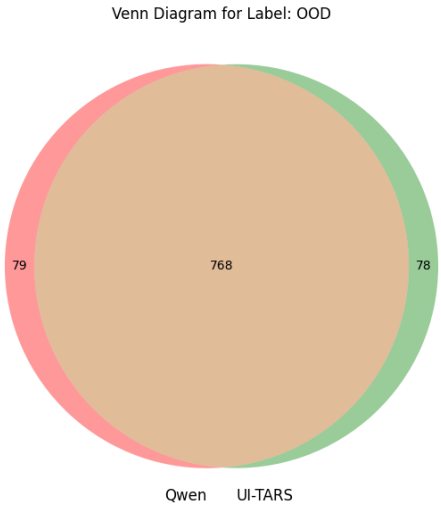



图 2: OOD

表 1: Qwen 模型下不同测试模型的表现（Type Match / Exact Match × ID / OOD）

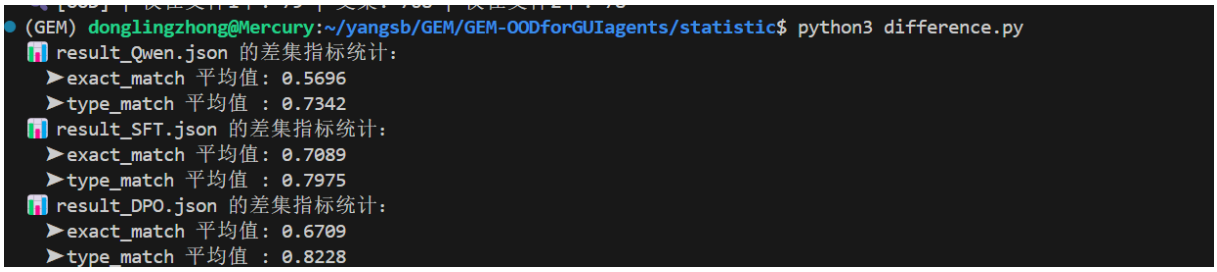
| 模型 | Type Match | | Exact Match | |
|---|---|---|---|---|
| | ID | OOD | ID | OOD |
| base | 0.6956409595047718 | 0.717827626918536 | 0.42661851947381996 | 0.4427390791027155 |
| SFT | 0.8047459375806035 | 0.8110979929161747 | 0.662625741552747 | 0.654073199527745 |
| RFT | 0.7913335052875935 | 0.8110979929161747 | 0.6463760639669848 | 0.6493506493506493 |

表 2: UI-TARS 模型下不同测试模型的表现（Type Match / Exact Match × ID / OOD）

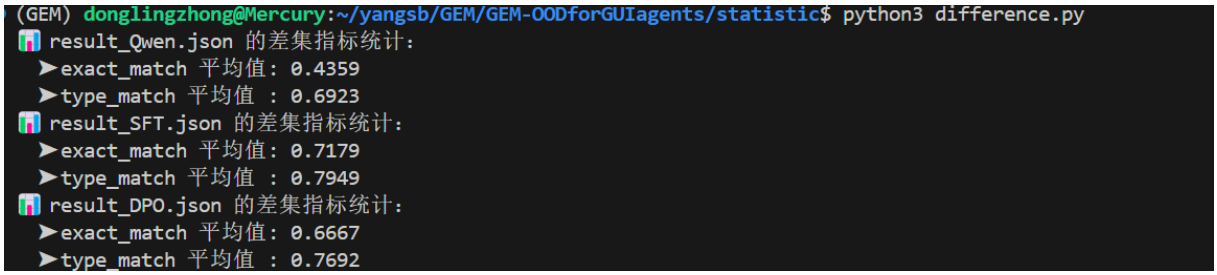
| 模型 | Type Match | | Exact Match | |
|---|---|---|---|---|
| | ID | OOD | ID | OOD |
| base | 0.6975244971634863 | 0.7092198581560284 | 0.427282104177411 | 0.4397163120567376 |
| SFT | 0.8047962867457452 | 0.8108747044917257 | 0.6619391438886024 | 0.6572104018912529 |
| RFT | 0.7921609076843734 | 0.8073286052009456 | 0.6467251160391955 | 0.6477541371158393 |

两个模型在 ID 上交集为 3799，Qwen 有 78 个额外的，UI-TARS 有 79 个额外的，在这 79 个上模型表现的均值，并将它们与对应模型下 ID 对应指标的均值进行对比：

UI-TARS 的 79 个：




可以观察到在这 79 个 ID 数据上 exact match: 0.57>0.43，0.71>0.66，0.67>0.65；type match: 0.73>0.70，0.7975<0.80，0.82>0.79，因此在这些数据上性能是有所提升的 Qwen 的 78 个：




可以观察到在这 78 个 ID 数据上 exact match: 0.436>0.426,0.7179>0.6626,0.6667>0.6463；type match: 0.6923<0.6956，0.7179<0.8047，0.7692<0.7913

# 3 实验结果分析

从单个模型的结果出发：SFT 和 RFT 都在 base 模型 ID 和 OOD 上的表现进行了提升，但是结果并没有表现出明显的 RFT 提升 OOD，SFT 提升 ID(来源于另一篇论文的观点，试图迁移得证)，不过从提升效果来看，type match 涨幅约 0.1，exact match 涨幅约 0.2

纵向比较 Qwen 和 UI-TARS 选作 GEM 划分使用的模型，两者从实验结果来看差距不大，说明模型本身的训练可能并不会带来 GEM 划分更优的情况 (与 GEM 论文中得出的结论一致，论文中有 UI-TARS-7B 和 Qwen-7B 结果的对比)，GEM 代码中 agent 起到的作用是提取 embedding，或许可以说 base 模型在经过 SFT、RFT 等方式的训练之后，提取 embedding 的能力并没有提升，或者专项训练的目的本来就不包括这一方面。

如上所示，我在 ID 差集的部分对均值进行了统计，试图在小样本数量的环境下得出 Qwen 和 UI-TARS 哪个模型的划分更为合理，从结果的角度来看在差集 ID 上，UI-TARS 在 type match 和 exact match 上都比 ID 均值更高 (5/6)，因此可以认为这些数据应该在 ID 中；而 Qwen 表现出反常 (3/6)，在 exact match 都是提升，但是在 type match 都是降低