上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

SJTU

Evaluating Code LLMs with 10% of the Data

董浚森

2026年1月7日

工作介绍

核心问题：评估大语言模型（LLM）能力的测试集过大，经常包括成千上万条。这就导致评测成本非常高，测试所需要的时间也非常长

发现：存在大量的样本冗余。

- 文本冗余：题目与其它的题目在语义上相似，考察能力相似
- 排名冗余：题目不同，但是在区分模型的能力上是重复的

目标：

使用尽可能少的样本预测原版全量测试的分数和水平。

——> 挑选的样本足够有代表性，可以充分反映整体目标

工作介绍

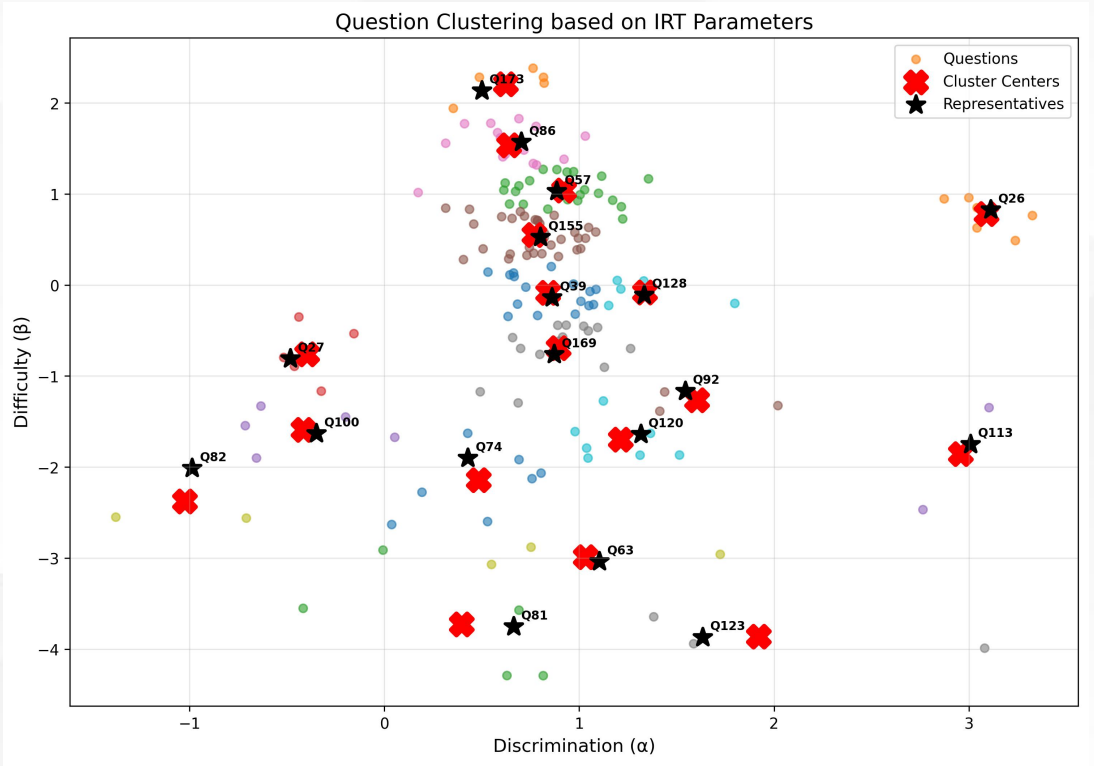
- 相关工作：tinyBenchmarks（最经典）
- 方法：

```
"q104": 1, "q105": 1, "q106": 1,
"q104": 0, "q105": 0, "q106": 0,
"q104": 0, "q105": 0, "q106": 0,
"q104": 0, "q105": 0, "q106": 1,
"q104": 1, "q105": 1, "q106": 1,
"q104": 1, "q105": 1, "q106": 1,
"q104": 1, "q105": 1, "q106": 0,
"q104": 1, "q105": 1, "q106": 0,
```

矩阵列出每一个模型在每一道题情况

IRT求解器
（项目反应理论）

求解出难度分布度



根据求解内容得到坐标，进行聚类选择锚点

实验配置

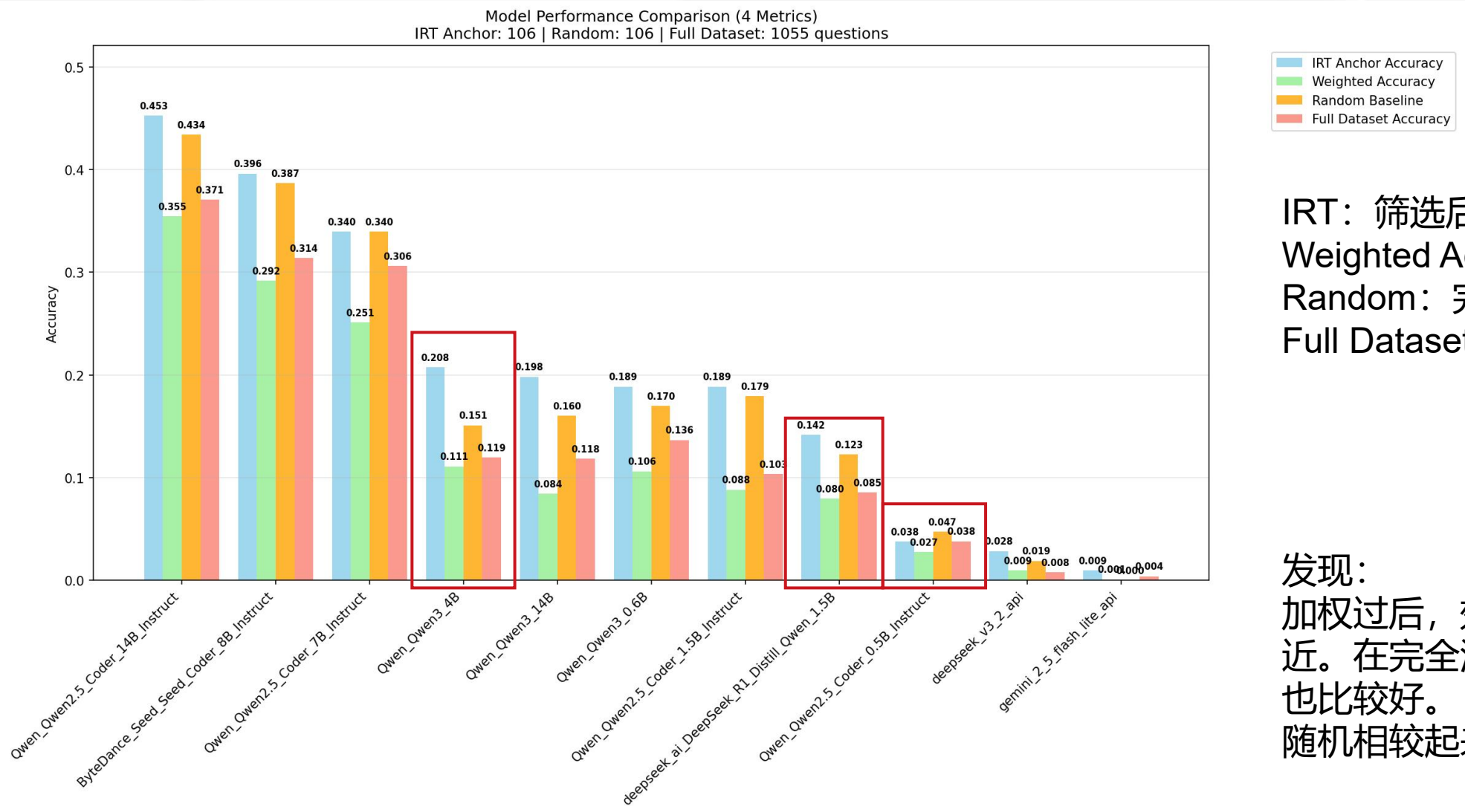
- 原始数据集：LiveCodeBench（用于严格评估大语言模型 (LLMs) 在代码处理方面的能力）
- 任务：让大模型去生成实现任务（生成/修复/执行/预测），评测是否完成
- 选用模型：

ByteDance_Seed_Seed_Coder_8B_Instruct
deepseek_ai_DeepSeek_R1_Distill_Qwen_1.5B
deepseek_ai_DeepSeek_R1_Distill_Qwen_7B
deepseek_v3_2_api
gemini_2_5_flash_lite_api
Qwen_Qwen2.5_Coder_0.5B_Instruct
Qwen_Qwen2.5_Coder_1.5B_Instruct
Qwen_Qwen2.5_Coder_7B_Instruct
Qwen_Qwen2.5_Coder_14B_Instruct
Qwen_Qwen3_0.6B
Qwen_Qwen3_4B
Qwen_Qwen3_14B

测试集任务：
随机选用11个模型中的8个作为基准模型，3个作为预测模型。测试在3个模型上的预测误差水平

测试集大小：
livecodebench 约1000道题目
按照时间顺序分为v1-v6（防止泄露），同时包含汇总全部的v7

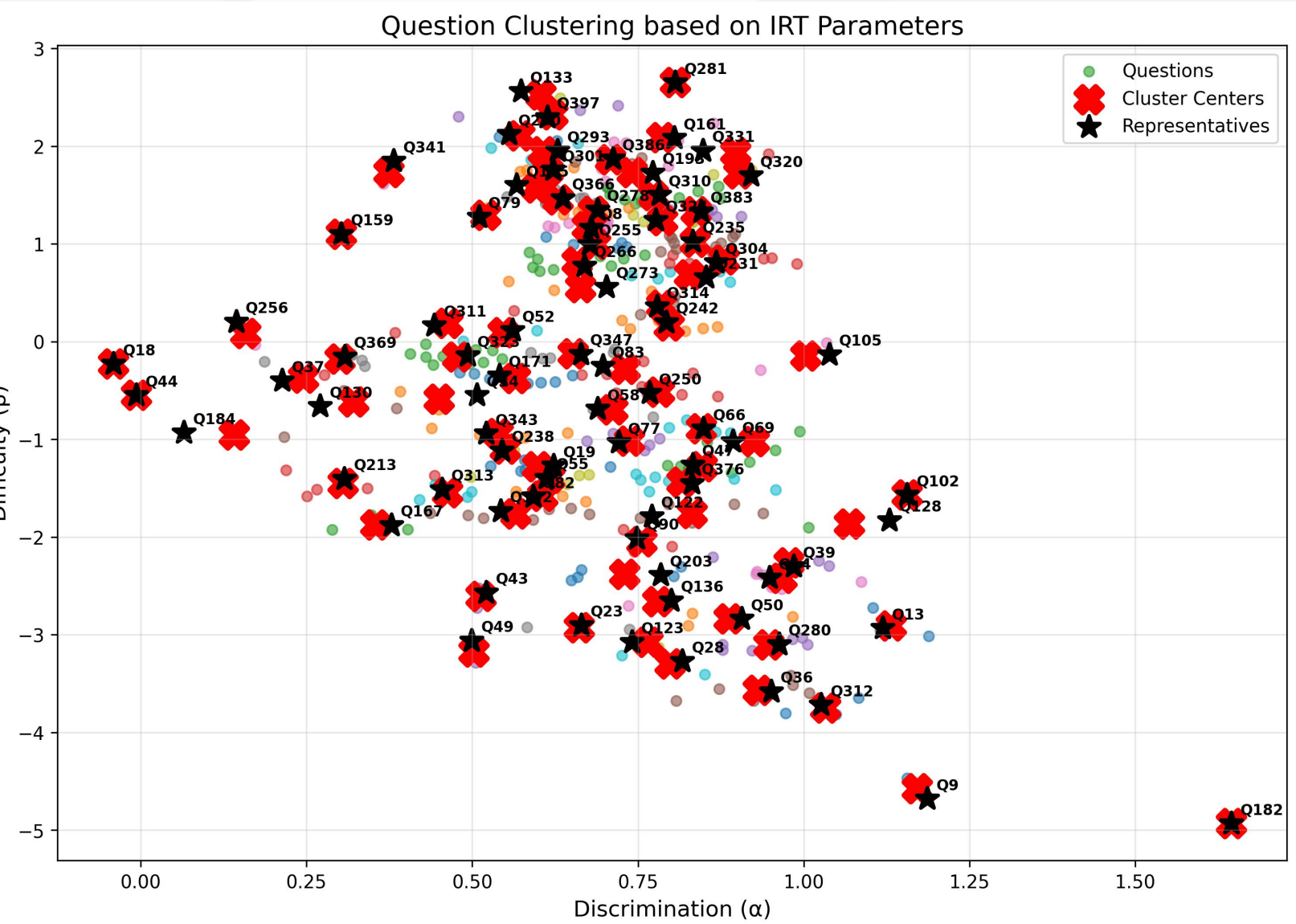
工作介绍



IRT：筛选后准确率
Weighted Accuracy：IRT加权准确率
Random：完全随机准确率
Full Dataset Accuray：全体准确率

发现：
加权过后，效果与原先结果非常接近。在完全没有接触的模型上表现也比较好。
随机相较起来偏差较大

工作介绍



选择的分布来看：

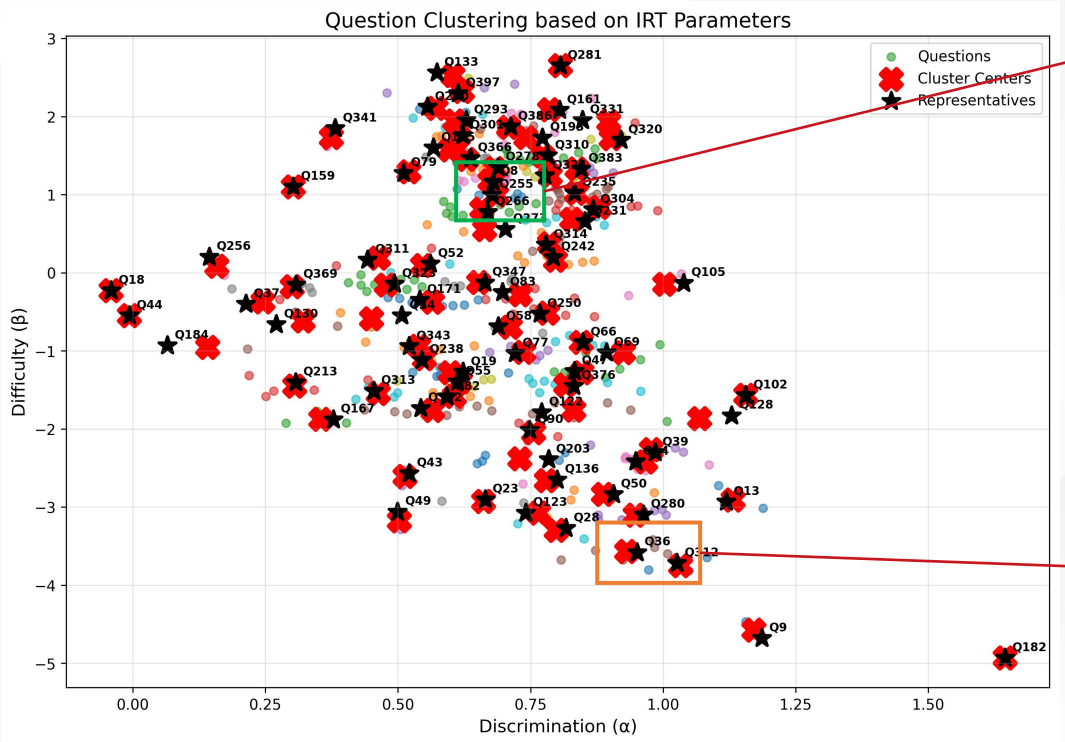
选择倾向保证各种类型的分布都有选择。

对于同一聚类，经过求解后，可以认为其分布和难度相似，属于同一种类型题目。

会优先的保证数据分布的“多样性”，同时，为了保证最终结果一致，又对这些不同的题目根据分析情况加权。最终保证加权准确率与原先一致，且准确率相对可靠。

数据分析

分析不同聚类的结果来看

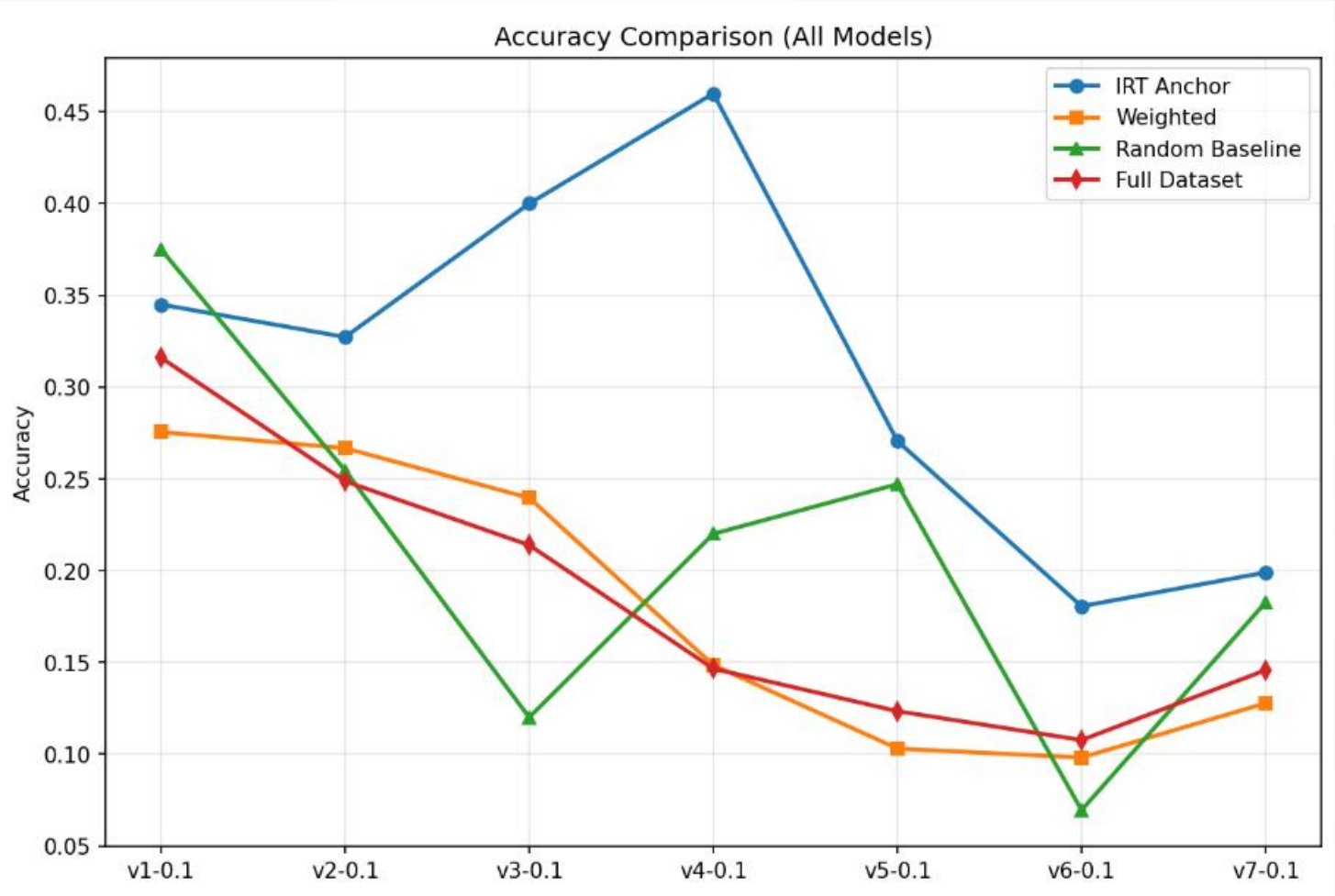


从中选取两个聚类分析特征

α（区分度） 中等，题目区分度比较大
β（难度） 比较高，难度大
题目分析：
题目整体水平比较难，需要比较高的代码水平才可以做这个题目，需要对题目进行比较复杂的建模和抽象

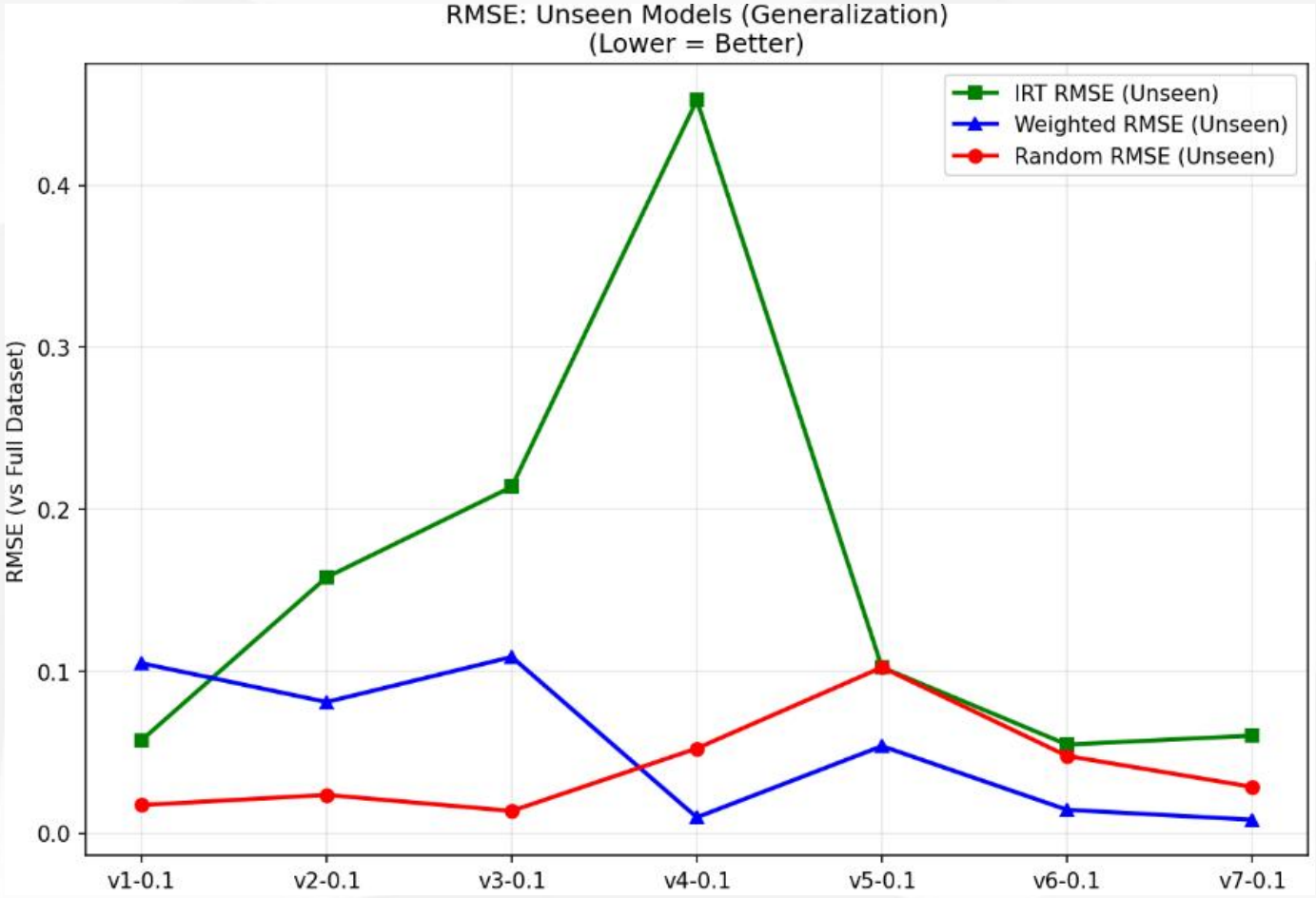
α 偏低，区分度中等偏高
β 偏低，题目较为简单
题目分析：
题目整体的水平很低但区分度很强，检查发现模型容易在一些细致末结处错误（边界，实现细节）

数据分析



- v1-v6是根据时间顺序的数据集，v1数据集最大，其余数据集比较小（100)。按照此方法筛选10%得到图
- v7是全体数据集合并为一个后取样得到的结果
- 结果：
- 筛选后结果与Full Dataset的结果比较接近，且摆动幅度相较于随机来看有明显进步，说明真的有筛选到高质量的数据

数据分析



$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

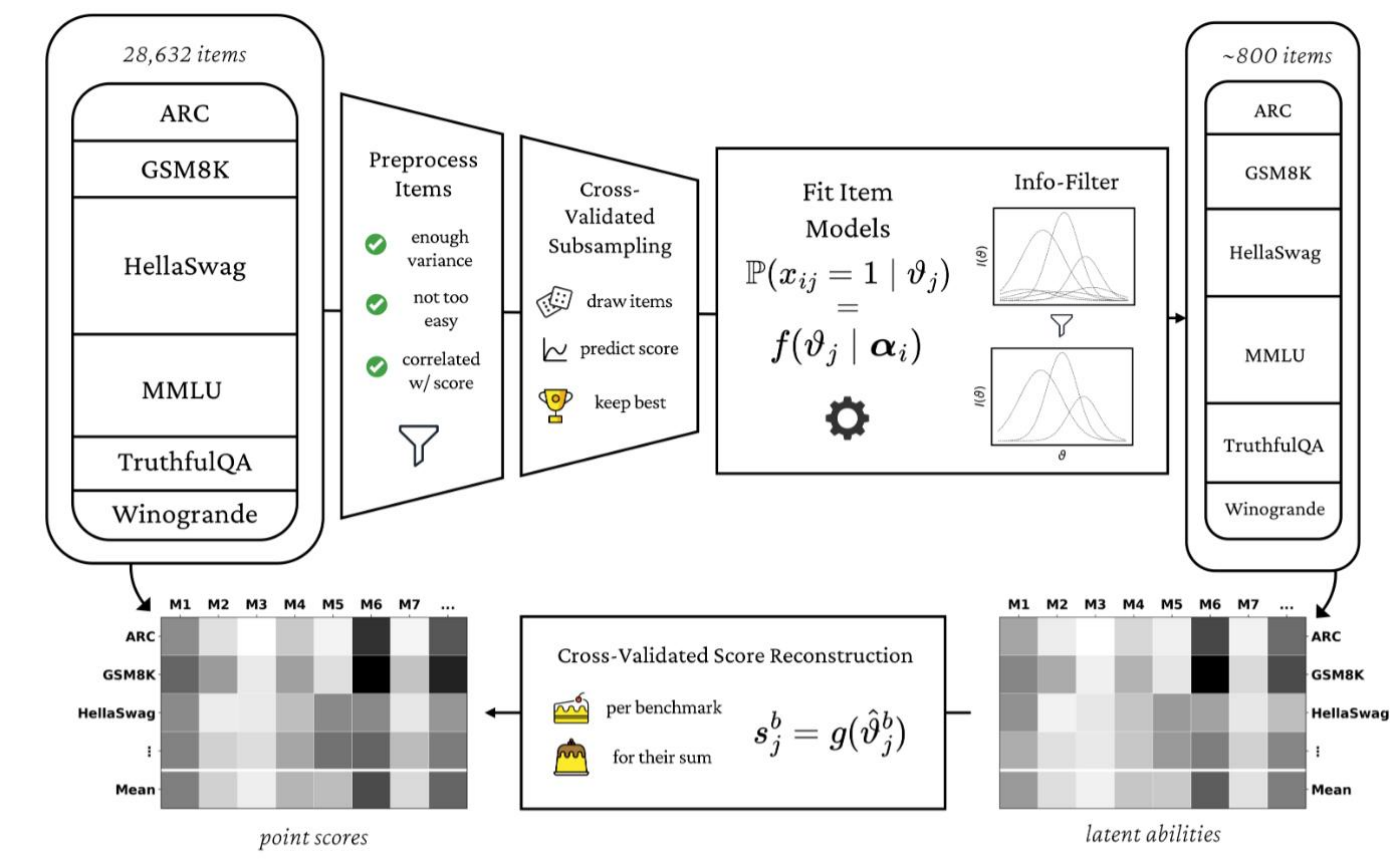
- 系统比较此方法的优势，采用RMSE，计算所有模型采样误差和真实值的RMSE，
- 可以看到，在数据量比较小的情况下，随机和筛选的效果可能互有胜负
- 但当数据量较大，可筛选子集比较大时候，误差显著低于随机取样

已有成熟方法汇总

已有方法汇总：

已有方法	方法		优势	模型数目
tinybenchmark	聚类	选取代表性的特征题目，加权得到总体水平	覆盖全面	37(小)/400(大)
metabench	费雪信息量	关心题目信息量	区分度极高	4000-6000(公开数据集)
essencebench	遗传算法	结果导向的暴力搜索最合适子集	结果导向、相关性强	4000-6000(公开数据集)
SubLIME	特征学习	依靠锚点模型和历史数据回归预测。	新榜单适应性强	313个模型10个基准测试；少量时优势明显

metabench



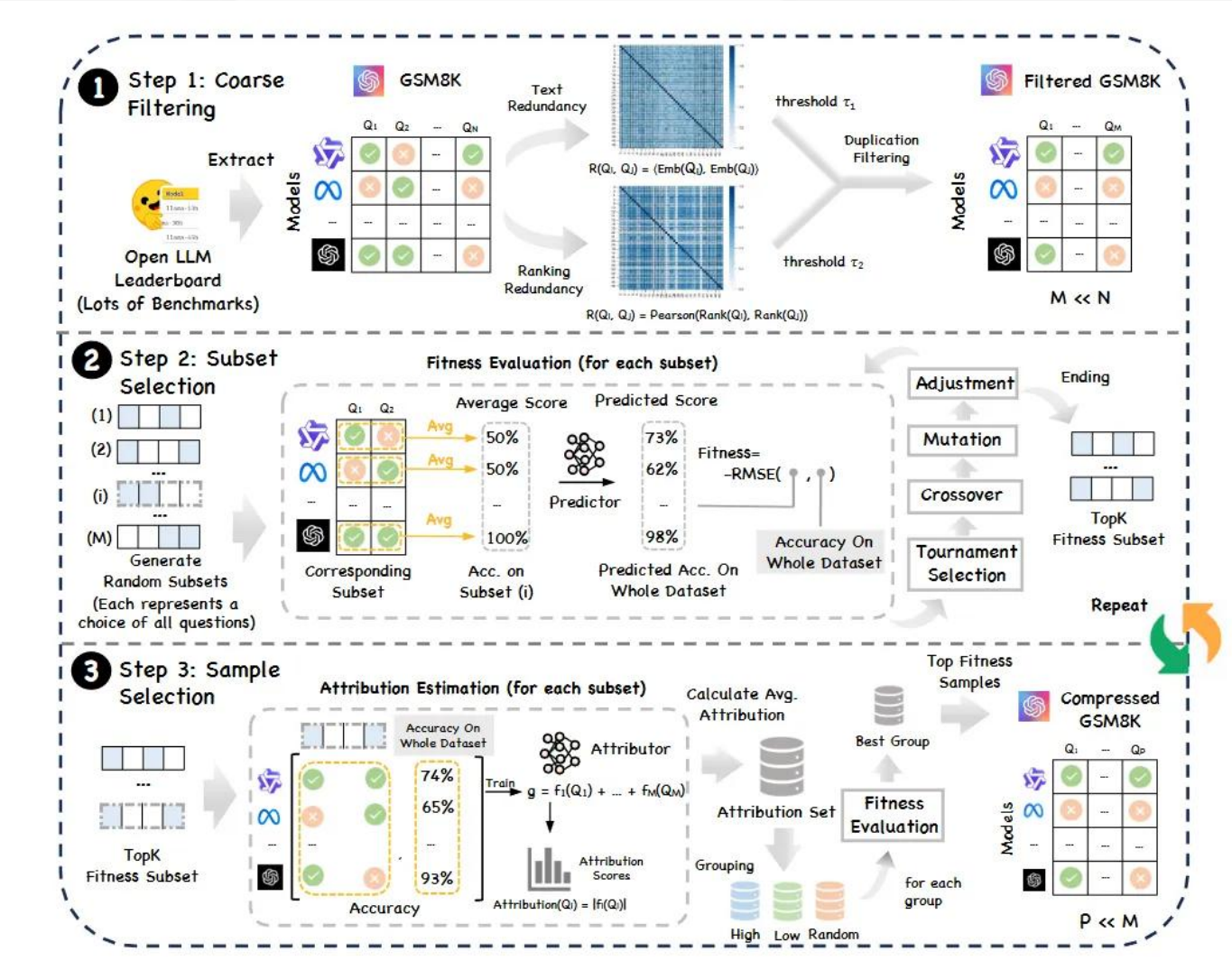
对基准数据进行初筛，去掉太简单太难的，方差过高的题目，保留高质量模型和题目。

建模时候，会给每一个模型潜在能力值和每一个题目的特征（难度区分度）

费雪信息量：根据每一个题目在不同能力水平下的信息量。选择出在每一个能力轴上面信息量最高的题目，最终组成。

最后进行回归预测，评判结果和分布是否保持一致。

essencebench



- 初筛：利用embadding相似度（语义）和排名相关性（题目难度与模型能力相关性），超过一定阈值的话剔除，只保留一个
- 子集筛选：使用**遗传算法**，选出特定的子集预测在样本上的得分，误差越小越好
- 为了避免陷入局部最优，计算样本贡献，再在组内精细化搜索得到结果

SubLime

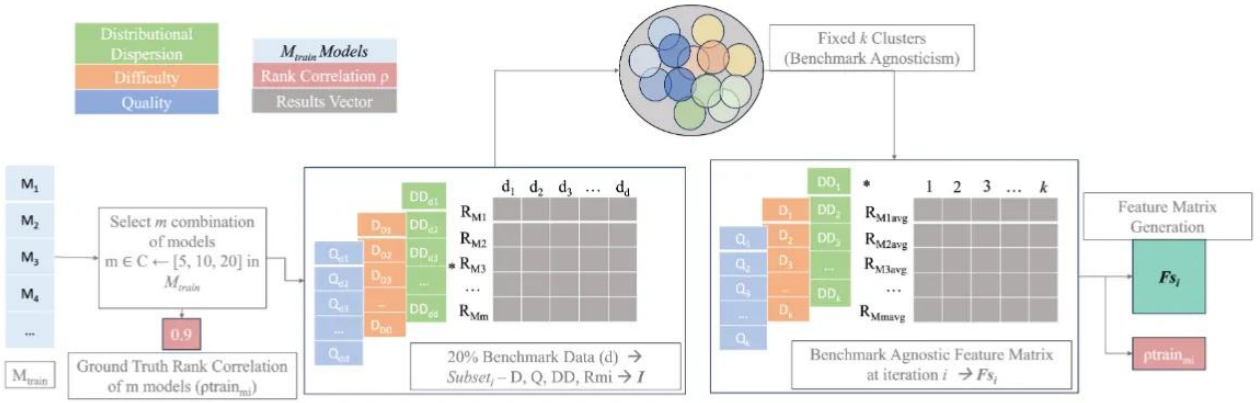
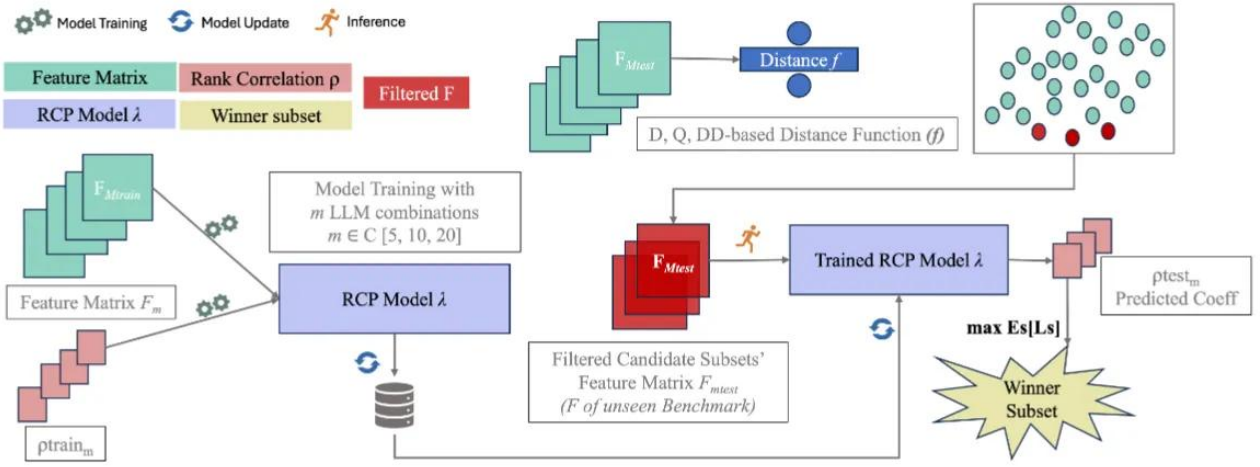


Figure 2: Feature Matrix F_{si} & Subset I_i Generation at Iteration i



- 对于数据集先得到固有指标D（难度），Q（质量），DD（离散度）
- 选取锚点模型，评估在候选子集上的结果，然后评测相关性。
- 训练一个模型来预测训练子集和整体数据的相关性。在需要评测新的Benchmark，只需要给锚点模型在生成的候选子集上进行少量评估，利用训练好的 RCP 模型预测并选出最能还原排名的“优胜子集”，即可用它代替全集来高效评测未来的新模型。
- D，Q参数依赖于同样是基于句子长度和音节数量的公式

思考

- 可能思考
- 1.最终目标：筛选出更好的数据集or更加符合原始分布的数据集。如果要更加符合分布则必然选入一些可能显然质量不高的题目。
- 2.筛选方法：问题本质是筛选子集仍然符合整体的性质
- 3.实验数据上：如训练借助现有公开数据集，本地模型数目有限

TODO

TODO：

- 1.尝试复现IRT++或其他论文，理论效果可能更佳
- 2.对livecodebench数据集筛选，选择更好的评测子集
- 3.继续进行casestudy

上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

感谢

饮 水 思 源　爱 国 荣 校