

PR #27457 完整报告

sgl-project/sglang

ci: show partition fit window as a date range in the step summary

合并时间: 2026-06-07 05:39

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/27457>

执行摘要

- 一句话: CI 分区窗口显示为日期范围
- 推荐动作: 小范围基础设施改进, 值得合并。建议后续考虑添加日期解析异常保护以增强健壮性。

功能与动机

之前分区步骤摘要仅打印 `data_as_of=2026-05-30...`, 这种表述看起来像是数据过期标记, 但实际上只是拟合窗口的起始日期 (比今天早 `fit_window_days` 天)。为了避免歧义, PR 改为展示完整的窗口范围 `[start, end)`, 让 CI 开发者直观了解模型训练数据覆盖的时间跨度。

实现拆解

1. 新增 `format_fit_window()` 辅助函数 (`scripts/ci/utils/compute_partitions.py`): 从模型字典中读取 `fit_window_start` 和 `fit_window_days`, 如果任一缺失则回退到简单显示 `fit_window_start` 值; 否则计算结束日期并返回格式化的字符串 `fit over [start, end)`。
2. 更新分区步骤摘要 (`scripts/ci/utils/compute_partitions.py`): 将原来硬编码的 `data_as_of` 字符串替换为 `format_fit_window(partition_model)` 调用, 使输出显示为 `live model: fit over [2026-05-30, 2026-06-06) (7d window)`。
3. 更新 `update_est_time.py` 诊断输出 (`scripts/ci/update_est_time.py`): 将 `stderr` 中的 `data_as_of` 替换为 `fit_window_start` 和 `fit_window_days`, 保持与上游数据源一致。
4. 移除 `data_as_of` 回退逻辑: 第三个 commit 删除了对旧键 `data_as_of` 的兼容性回退, 因为上游重命名已经生效, 无需双读。

关键文件:

- `scripts/ci/utils/compute_partitions.py` (模块 CI 脚本; 类别 infra; 类型 infrastructure; 符号 `format_fit_window`): 新增 `format_fit_window` 函数并更新分区摘要行, 是变更核心文件。
- `scripts/ci/update_est_time.py` (模块 CI 脚本; 类别 infra; 类型 infrastructure): 更新 `stderr` 诊断输出, 从 `data_as_of` 改为 `fit_window_start` 和 `fit_window_days`, 保持一致性。

关键符号: `format_fit_window`

关键源码片段

scripts/ci/utis/compute_partitions.py

新增 `format_fit_window` 函数并更新分区摘要行，是变更核心文件。

```
# scripts/ci/utis/compute_partitions.py

def format_fit_window(model: dict) -> str:
    """Render the model's fit window as `[start, end)` for step summary.

    Surfacing the whole span keeps the lower-bound date from reading as a
    staleness marker (it trails today by fit_window_days).
    """
    start = model.get("fit_window_start")
    days = model.get("fit_window_days")
    if not start or not isinstance(days, int):
        return f"fit_window_start={start}"
    # 计算结束日期: start + days
    end = (datetime.strptime(start[:10], "%Y-%m-%d") + timedelta(days=days)).strftime(
        "%Y-%m-%d"
    )
    return f"fit over [{start[:10]}, {end}) ({days}d window)"

# 在 main() 中使用
# 原 : f"live model `data_as_of={partition_model.get('data_as_of')}`", "
# 新 : f"live model: {format_fit_window(partition_model)}, "
```

scripts/ci/update_est_time.py

更新 `stderr` 诊断输出，从 `data_as_of` 改为 `fit_window_start` 和 `fit_window_days`，保持一致性。

```
# scripts/ci/update_est_time.py
# 原 : print(f" model data_as_of={model.get('data_as_of')} ", file=sys.stderr)
# 新 :
print(
    f" model fit_window_start={model.get('fit_window_start')} "
    f"fit_window_days={model.get('fit_window_days')} "
    f"n_runs={model.get('n_runs')} "
    f"n_suites={len(model.get('est', {}))}",
    file=sys.stderr,
)
```

评论区精华

审查者 `gemini-code-assist[bot]` 建议在 `format_fit_window` 函数中对日期解析部分添加 `try-except` 块，以处理意外或格式错误的日期字符串，防止 CI 分区崩溃。作者未采纳该建议（未回复且未修改），推测认为上游数据源可信度高，添加异常处理的收益有限。

- 添加日期解析异常保护 (`correctness`): 作者未采纳，未修改。认为上游数据格式稳定，且回退路径已存在。

风险与影响

- 风险：风险较低。主要风险是上游数据源（sglang-ci-stats）尚未完全迁移到 `fit_window_start` 键名时，`format_fit_window` 函数会因缺少该键而回退到显示原始值，但不会崩溃。由于 PR 已移除对旧键 `data_as_of` 的回退，如果上游回滚，CI 摘要可能显示不完整信息。另外，日期解析未加异常保护，若上游数据结构意外变化，`datetime.strptime` 可能抛出异常导致步骤失败，但概率极低。
- 影响：影响范围：仅限 `scripts/ci` 模块，影响 CI 分区步骤的日志输出和调试诊断。对运行时的模型推理、性能等无任何影响。CI 开发者在查看分区摘要时可以更直观地理解模型训练窗口，提升可读性。
- 风险标记：日志错误处理缺失，依赖外部数据源

关联脉络

- PR #5 Rename model.json `data_as_of` to `fit_window_start`: 此 PR 依赖的键名重命名来自该 issue，是变更的数据上游。