

PR #2476 完整报告

radixark/miles

fix/dashboard): zero the trainer log-probs the loss masks out

合并时间: 2026-08-13 01:53

原文链接: <http://prhub.com.cn/radixark/miles/pull/2476>

执行摘要

- 一句话: 清零 loss 掩码位置的 train log-probs, 修复 token 视图误导
- 推荐动作: 值得精读。本 PR 虽然改动很小, 但解决了一个典型的“占位值污染可视化”问题, 展示了如何区分“真实零值”与“缺失占位”, 并通过保持数组对齐的方式在展示层修复而不影响统计层。_zero_masked 的设计简洁可复用, 是 dashboard 数据展示层的一个好范例。

功能与动机

PR body 明确指出: dump 对 loss 忽略的位置没有 rollout log-prob (引擎从未生成这些 token, 所以存的是 0.0 占位), 而 trainer 会给每个位置打分, token 视图直接相减会把真实 log-prob 与占位符比较, 在从不训练的 token 上产生 tens of nats 的虚假分歧。更严重的是 `lp_diff` 热力图是发散色标、按窗口内最大绝对值归一化, 单个被掩码的 -30 会把所有真实值压成中性色, 导致用于定位 drift 的视图完全失效。

实现拆解

实现分为 4 步:

1. 新增掩码清零辅助函数: 在 `miles/dashboard/dump_reader.py` 中新增 `_zero_masked(values, mask)`, 用 `torch.where(mask, values, zeros_like(values))` 将掩码位置清零但保留数组长度, 确保与 token 切片一一对齐。
2. 改造 `tokens()` 计算链路: 在 `DumpReader.tokens()` 中先用 `_zero_masked(row.log_probs, row.loss_mask > 0)` 得到 `train_log_probs`, 再基于它计算 `lp_diff` (`train_log_probs - row.rollout_log_probs`), 最终返回给前端的 `train_log_probs`、`lp_diff`、`imp_ratio` 都使用这条新链路。
3. 补充回归测试: 在 `tests/fast/dashboard/test_dump_reader_views.py` 新增 `test_tokens_blank_the_trainer_side_where_the_loss_is_masked`, 覆盖全掩码样本 (REMOVED[0]) 与普通样本, 断言掩码位置的 `train_log_probs` 为 0、`lp_diff` 等于 `-rollout_log_probs`、`ref_log_probs` 保持原样, 且未掩码样本的值不变。
4. 同步文档: 在 `docs/user-guide/dashboard.md` 的 token 页面说明中新增注意事项, 告知用户 loss 忽略的位置两边都读为零, 应只依据 loss 覆盖的 token 判断一致性。

该改动刻意只作用于 token 视图, 汇总统计量 (`mean_abs_lp_diff`、`max_abs_lp_diff`、`mean_imp_ratio`) 此前已只选择 loss 覆盖的位置, 因此不受影响; `ref_log_probs` 也保持 dump 原样。

关键文件：

- miles/dashboard/dump_reader.py（模块 数据读取；类别 source；类型 core-logic；符号 `_zero_masked`, `tokens`）：核心源码变更：新增 `_zero_masked` 辅助函数并改造 `tokens()` 的 `train_log_probs/lp_diff` 计算链路，是修复的载体。
- tests/fast/dashboard/test_dump_reader_views.py（模块 视图测试；类别 test；类型 test-coverage；符号 `test_tokens_blank_the_trainer_side_where_the_loss_is_masked`）：新增回归测试覆盖全掩码与普通样本，验证清零逻辑及与 rollout 侧占位值的关系。
- docs/user-guide/dashboard.md（模块 用户文档；类别 docs；类型 documentation）：同步说明 token 页面中 loss 忽略位置的展示约定，避免用户误读。

关键符号：`_zero_masked`, `tokens`, `test_tokens_blank_the_trainer_side_where_the_loss_is_masked`

关键源码片段

miles/dashboard/dump_reader.py

核心源码变更：新增 `_zero_masked` 辅助函数并改造 `tokens()` 的 `train_log_probs/lp_diff` 计算链路，是修复的载体。

```
# miles/dashboard/dump_reader.py
```

```
def _zero_masked(values: torch.Tensor | None, mask: torch.Tensor) -> torch.Tensor | None:
```

```
    """将 loss 掩码位置清零，同时保持数组长度与 token 切片对齐。
```

```
    token 视图需要保留每一个位置（数组要与 token 一一对应），  
    只有汇总统计量才可以真正丢弃被掩码的位置。
```

```
    """
```

```
    # None 表示该行没有 trainer 数据（如 evaluation 样本），直接透传。
```

```
    return None if values is None else torch.where(mask, values, torch.zeros_like(values))
```

```
# 在 DumpReader.tokens() 中，改造后的核心计算链路：
```

```
#
```

```
# dump 对 loss 忽略的位置（工具输出、被掩码的 turn）没有 rollout log-prob:
```

```
# 引擎从未生成这些 token，因此存的是 0.0 占位。trainer 却会给每个位置打分，
```

```
# 若直接相减，lp_diff 会在从不参与训练的 token 上报告几十 nats 的差异，
```

```
# 并在分发热力图中把真实信号压平。这里把 trainer 侧同样清零。
```

```
train_log_probs = None if row is None else _zero_masked(row.log_probs, row.loss_mask > 0)
```

```
lp_diff = (
```

```
    train_log_probs - row.rollout_log_probs
```

```
    if row is not None and train_log_probs is not None and row.rollout_log_probs is not None
```

```
    else None
```

```
)
```

```
# 之后 token 视图返回的 train_log_probs、lp_diff、imp_ratio 都基于上述结果。
```

评论区精华

本 PR 没有公开的 review 讨论线程，Zhichenzzz 直接批准（APPROVED，无评论）。PR body 中作者自行交代了设计取舍：清零只限于 token 视图，因为该视图需要保留每个位置以保持数组对齐；汇总统计量早已按 loss 掩码过滤；`ref_log_probs` 暂时不做同样处理，因为当前没有视图对它做差分，未来若增加参考模型对比视图需要同样处理。

- 暂无高价值评论线程

风险与影响

- 风险：

1. tensor 类型假设：`_zero_masked` 依赖 `torch.where`，若 `row.log_probs` 不是 tensor（例如未来 dump 格式变化），会直接报错；当前 `TrainRow` 数据来自 `torch.load`，风险较低。
2. 热力图行为变化：掩码位置全部变为 0 后，`lp_diff` 热力图的色标窗口会缩小到真实信号的范围，原本“被异常值压平”的症状消失，但用户若未读文档，可能误以为掩码位置“没有数据”；文档已补充说明。
3. 测试覆盖缺口：作者明确说明本地无法运行 dashboard 测试（`sglang/triton` 依赖在 macOS 上不可用），新测试依赖 CI 验证；PR 已合入，CI 结果未在材料中给出，存在理论上的回归风险。
4. 与其他视图的一致性：`ref_log_probs` 未同步处理，若未来新增“参考模型 vs trainer”的差分视图，会重蹈本 PR 修复的误导问题。
 - 影响：用户影响：dashboard 的 token 页面不再显示虚假的几十 nats 分歧，`lp_diff` 热力图能真实反映 loss 覆盖位置的训练 / 推理差异，避免用户误判 run 健康状态。系统影响：仅增加一次 `torch.where` 操作，无性能与存储影响。团队影响：文档同步更新，降低了 dashboard 解读门槛；新增 `_zero_masked` 成为可复用的掩码处理工具，为后续视图开发提供约定。
 - 风险标记：热力图尺度行为变化，新测试依赖 CI 验证，`ref_log_probs` 未同步处理

关联脉络

- PR #2027 dashboard: advisory v2 — run-health alarms before config tuning: 同属 dashboard 数据解读与健康告警改进，都在完善如何准确判断 run 健康状态，避免被视图中的假象误导。
- PR #2353 dashboard: report model FLOPs utilization: 同属 dashboard 观测指标增强，涉及 `dump_reader.py` 及 dashboard 文档，与本 PR 共同演进 dashboard 的指标展示正确性。