

PR #1984 完整报告

THUDM/slime

rename rollout_ids to group_ids

合并时间: 2026-05-30 08:55

原文链接: <http://prhub.com.cn/THUDM/slime/pull/1984>

执行摘要

- 一句话: 将 rollout_id 重命名为 group_id, 统一聚合单元概念
- 推荐动作: 值得精读。该 PR 展示了如何在不破坏外部写入的情况下安全废弃一个字段, `__getattribute__`/`__setattr__` 的用法是很好的实战案例。对于维护核心数据结构的团队有参考价值。

功能与动机

原有 `rollout_id` 命名容易让人误解为一次 rollout 执行的唯一标识, 但实际上该 ID 用于 loss 聚合: 同一 rollout 发出的多个样本共享相同 ID 以避免 loss 过计。重命名为 `group_id` 能更清晰地传达“训练聚合单元”这一语义, 减少理解成本。PR 说明虽未给出显式动机, 但从 diff 中的注释和文档更新可清楚看到该意图。

实现拆解

1. 核心数据结构变更 (`slime/utils/types.py`): 在 `Sample` dataclass 中新增 `group_id` 字段替代 `rollout_id`, 并覆盖 `__getattribute__` 和 `__setattr__` 实现读废弃写兼容的过渡策略——读取 `rollout_id` 抛出 `AttributeError`, 写入时转发到 `group_id` 并发出 `DeprecationWarning`。 `from_dict` 同时接收旧键 `rollout_id` 以兼容历史序列化数据。
2. rollout 主路径适配 (`slime/ray/rollout.py`): `_get_rollout_data` 中验证函数从 `_validate_rollout_id_annotated` 替换为 `_validate_group_id_annotated`, 注释同步更新。`_convert_samples_to_train_data` 中训练数据字典键名从 `rollout_ids` 变为 `group_ids`, 取值逻辑从三元条件简化为 `sample.group_id if sample.group_id is not None else sample.index`。
3. DP 调度器参数重命名 (`slime/utils/dp_schedule.py`): `build_dp_schedule` 函数参数 `rollout_indices` 重命名为 `group_indices`, 所有内联注释和变量名同步更新。调度的基本单元从“rollout”改为“group”。
4. Megatron 后端适配 (`slime/backends/megatron_utils/data.py`, `actor.py`, `loss.py`): 训练数据键名 `rollout_ids` → `group_ids`、`rollout_mask_sums` → `group_mask_sums`, 日志和 loss 计算中的注释同步调整。
5. Agent 及其他模块 (`slime/agent/trajectory.py`, `examples/multi_agent/agent_system.py`, `slime/rollout/forge_load.py` 等): 将 `rollout_id` 引用改为 `group_id`, 保持接口一致。

6. 测试与文档: `tests/test_sample.py` 新增四个测试覆盖向后兼容场景 (仅写、旧键读取异常、`from_dict` 优先 `group_id`、序列化不输出废弃键) ; `tests/test_dp_schedule.py` 同步更新参数名和断言; `docs` 目录下相关文档同步更新术语。

关键文件:

- `slime/ray/rollout.py` (模块 Rollout; 类别 source; 类型 core-logic; 符号 `_validate_rollout_id_annotated`, `_validate_group_id_annotated`) : 核心 rollout 主路径, 验证和训练数据生成中的 `rollout_id` 改为 `group_id`, 是本次重命名的关键使用方。
- `slime/utils/types.py` (模块 类型系统; 类别 source; 类型 dependency-wiring; 符号 `getattr`, `setattr`) : 核心数据类型 `Sample` 的字段重命名和废弃机制实现, 是本次变更的源头。
- `tests/test_sample.py` (模块 单元测试; 类别 test; 类型 test-coverage; 符号 `test_group_id_accepts_legacy_rollout_id_assignment_only`, `test_from_dict_accepts_legacy_rollout_id_without_group_id`, `test_from_dict_prefers_group_id_over_legacy_rollout_id`, `test_group_id_does_not_serialize_legacy_rollout_id_alias`) : 新增四个测试全面覆盖向后兼容场景, 确保旧写入、旧读取异常、`from_dict` 优先级、序列化行为正确。
- `tests/test_dp_schedule.py` (模块 调度测试; 类别 test; 类型 test-coverage; 符号 `test_rollout_grouping_keeps_samples_together`, `test_grouping_keeps_samples_together`, `test_trims_trailing_rollouts_that_dont_fill_a_step`, `test_trims_trailing_groups_that_dont_fill_a_step`) : 同步更新 DP 调度测试中的参数名和断言, 验证调度逻辑随重命名正常工作。
- `slime/utils/dp_schedule.py` (模块 DP 调度; 类别 source; 类型 core-logic) : DP 调度器的核心函数参数名重命名, 调度单元从 `rollout` 改为 `group`。

关键符号: `Sample.getattribute`, `Sample.setattr`, `Rollout._convert_samples_to_train_data`, `build_dp_schedule`, `log_rollout_data`

关键源码片段

`slime/ray/rollout.py`

核心 rollout 主路径, 验证和训练数据生成中的 `rollout_id` 改为 `group_id`, 是本次重命名的关键使用方。

```
def _convert_samples_to_train_data(self, samples: list[Sample] | list[list[Sample]]):
    # ... 前面的奖励处理 ...
    raw_rewards, rewards = self._post_process_rewards(samples)

    # 每个 group 一个 ID (训练聚合单元)。普通 rollout 每个 group 只有一个样本,
    # 因此回退到样本的 index。compact / subagent 路径会显式设置 group_id
    # 以便兄弟样本共享同一 ID, loss 聚合时只计一次。
    group_ids = [
        sample.group_id if sample.group_id is not None else sample.index
        for sample in samples
    ]
```

```

train_data = {
    "tokens": [sample.tokens for sample in samples],
    # ... 其他字段 ...
    "group_ids": group_ids, # 旧名 rollout_ids → 新名 group_ids
}
# ... 后续 loss masks 等处理 ...

```

slime/utils/types.py

核心数据类型 **Sample** 的字段重命名和废弃机制实现，是本次变更的源头。

```

# Sample 类新增的魔法方法 __getattr__ 和 __setattr__
def __getattr__(self, name):
    if name == "rollout_id":
        raise AttributeError(
            "Sample.rollout_id is deprecated and write-only; use Sample.group_id instead."
        )
    return object.__getattr__(self, name)

def __setattr__(self, name, value):
    if name == "group_id":
        object.__setattr__(self, "group_id", value)
        return
    if name == "rollout_id":
        # 写兼容：将旧键 rollout_id 赋值转发到 group_id，并发出弃用警告
        if value is None:
            return
        warnings.warn(
            "Sample.rollout_id is deprecated and write-only; set Sample.group_id instead.",
            DeprecationWarning,
            stacklevel=2,
        )
        object.__setattr__(self, "group_id", value)
        return
    object.__setattr__(self, name, value)

# from_dict 中兼容旧键的处理（部分）
@staticmethod
def from_dict(data: dict):
    # ... 原处理 ...
    for key, value in data.items():
        if key not in field_names:
            # 兼容旧版 rollout_id: 仅当 group_id 未设置时才使用旧值
            if key == "rollout_id":
                if sample.group_id is None:
                    setattr(sample, key, value)
                    continue
            setattr(sample, key, value)
    return sample

```

tests/test_sample.py

新增四个测试全面覆盖向后兼容场景，确保旧写入、旧读取异常、from_dict 优先级、序列化行为正确。

```
@pytest.mark.unit
def test_group_id_accepts_legacy_rollout_id_assignment_only():
    """Older custom rollout code may assign `rollout_id`; constructor use is no longer supported.
    """
    # 构造阶段传入 rollout_id 应直接报错（类型不匹配，因为 dataclass 已无该字段）
    with pytest.raises(TypeError, match="rollout_id"):
        Sample(index=42, rollout_id=9)

    # 赋值阶段写入 rollout_id 应被转发到 group_id，并抛出弃用警告
    sample = Sample(index=42)
    with pytest.warns(DeprecationWarning, match="Sample.rollout_id is deprecated"):
        sample.rollout_id = 10
    assert sample.group_id == 10
    assert sample.to_dict()["group_id"] == 10

    # 读取 rollout_id 应触发写专属（write-only）错误
    with pytest.raises(AttributeError, match="write-only"):
        _ = sample.rollout_id
```

评论区精华

本 PR 无 review 评论和讨论。从 diff 可知作者独立完成了自审和合并。

- 暂无高价值评论线程

风险与影响

- 风险：
 1. 向后兼容风险（主要）：Sample.rollout_id 字段被废弃，任何尝试读取 sample.rollout_id 的代码将抛出 AttributeError。外部自定义 rollout 代码若依赖该读取，需修改为 sample.group_id。写入路径虽保留，但会触发警告，可能破坏静默升级场景。
 2. 序列化兼容风险：历史持久化的 rollout 数据（如 debug dump）中字段名为 rollout_id，新版 from_dict 虽能兼容，但 to_dict 不再输出它，导致回滚旧版时可能丢失 ID。
 3. 配置参数名变更：build_dp_schedule 的参数和部分内部变量名变更，外部直接调用该函数的代码（如有）需同步更新。- 影响：影响范围：全代码库 20 个文件，涉及核心数据结构 Sample、rollout 调度器、Megatron 训练后端、agent 模块和多个示例。所有使用 rollout_id 字段或 rollout_ids 键名的代码都必须适配。影响程度中等，因为变更主要是重命名，逻辑等价，但弃用策略要求调用方及时迁移。- 风险标记：向后兼容风险，核心字段废弃，序列化兼容隐患，多模块协同变更

关联脉络

- PR #1965 Don't use sample.index as default rollout_id: 该 PR 修正了默认 rollout_id 的行为, 本次重命名进一步巩固了 group_id 作为聚合标识的一致性。
- PR #1960 Extract more util code from coding_agent_rl example: 该 PR 重构了 agent 工具, 其中也涉及 rollout_id 的处理, 本次重命名需要同步更新 trajectory.py。