PR #2241 完整报告

THUDM/slime

fix: restore negative dataset slice bounds (path@[-100:])

合并时间：2026-08-12 13:32

原文链接：http://prhub.com.cn/THUDM/slime/pull/2241

执行摘要

- 一句话：修复数据集路径负索引切片崩溃
- 推荐动作：该 PR 值得快速阅读，但除非团队经常使用负索引切片，否则不必精读。值得关注的设计决策是「按边界符号分流：非负走流式 islice、负边界物化走列表切片」，这是一个务实的兼容性取舍。建议阅读 slime/utils/data.py 的 read_file 修改和新增测试文件，理解其权衡过程；后续如有大规模数据集使用负切片的场景，可考虑更优方案（如先统计行数再转正索引）。

功能与动机

PR body 明确说明：--prompt-data data.jsonl@[-100:]（取最后 100 行）会崩溃并抛出 ValueError，尽管 slice 解析器显式接受负边界。该语法最初以 pandas 语义（df.iloc[row_slice]）支持负索引，但在 #696 的流式重写中改用 itertools.islice，导致任何负边界都触发 ValueError: Indices for islice() must be None or an integer: 0 <= x <= sys.maxsize。解析器正则 (?P<start>-?\d*):(?P<end>-?\d*) 仍对两个边界显式接受符号，因此功能被宣传但实际不可用。

实现拆解

该 PR 实现分为以下几步：

1. 修改核心读取逻辑（slime/utils/data.py）：在 read_file 中，当 row_slice 存在时，检测 start 或 stop 是否为负值。若为负值，则将 reader 物化为 list(reader) 并应用 Python 原生列表切片（list(reader)[row_slice]），再包装为迭代器；否则保持原有 itertools.islice 流式路径不变。原因：解析负边界需要知道总行数，只能物化；而物化正是 #696 之前 pandas 实现的行为，因此不会引入额外内存回归。
2. 新增单元测试（tests/test_read_file_slicing.py）：创建了 82 行的测试文件，覆盖路径解析、无切片读取全部、非负切片（@[0:3]、@[3:]、@[:4]）、负切片（@[-3:]、@[:-2]、@[1:-1]、@[-5:-2]）以及负边界大于文件行数的情况（@[-100:] 在 10 行文件上应返回全量）。
3. 注册 CI 测试任务（.github/workflows/pr-test.yml 与 pr-test.yml.j2）：将新测试文件 test_read_file_slicing.py 加入 cpu-unittest 作业的测试列表中，确保该测试在 PR 验证时自动运行。

关键文件：

- slime/utils/data.py（模块 数据读取；类别 source；类型 core-logic）：核心逻辑修复文件。read_file 中负边界切片从 itertools.islice 改为物化列表切片，是本次变更的核心。
- tests/test_read_file_slicing.py（模块 测试；类别 test；类型 test-coverage；符号 jsonl_path, _ids, test_parse_generalized_path, test_no_slice_reads_everything）：新增 82 行单元测试，覆盖解析、非负切片、负切片及负边界大于文件等场景，是本次修复的回归保护。
- .github/workflows/pr-test.yml（模块 CI 配置；类别 infra；类型 infrastructure）：将新增测试文件注册到 cpu-unittest 作业，确保 CI 覆盖。
- .github/workflows/pr-test.yml.j2（模块 CI 模板；类别 infra；类型 infrastructure）：CI 配置模板同步更新，保证模板生成的 workflow 包含新测试。

关键符号：read_file, _parse_generalized_path

关键源码片段

slime/utils/data.py

核心逻辑修复文件。read_file 中负边界切片从 itertools.islice 改为物化列表切片，是本次变更的核心。

```
# slime/utils/data.py — read_file 中的切片应用逻辑（修复后）

def read_file(path, row_slice=None):
    # ... 前面根据文件后缀构造 reader（jsonl_reader / parquet_reader）...

    if row_slice is not None:
        logger.info("read_file path=%s applying slice row_slice=%s", path, row_slice)

        # islice 不允许负索引，但 @[...] 语法支持负边界（如 @[-100:] 表示取最后 100 行）。
        # 解析负边界需要知道总行数，所以该场景下把 reader 物化为列表再切片；
        # 非负边界保持流式 islice，避免不必要的内存开销。
        if (row_slice.start or 0) < 0 or (row_slice.stop or 0) < 0:
            # 物化后直接用 Python 列表切片，语义与 list[row_slice] 完全一致
            reader = iter(list(reader)[row_slice])
        else:
            # 更常见的场景：流式读取，避免把整个文件加载进内存
            reader = itertools.islice(reader, row_slice.start, row_slice.stop, row_slice.step)

    yield from reader


def _parse_generalized_path(s: str):
    # 正则显式允许 start/end 带符号，例如 "data.jsonl@[-100:]"
    if (m := re.match(r"^(?P<real_path>.*)@\[(?P<start>-?\d*):(?P<end>-?\d*)\]$", s)) is not None:
        path = m.group("real_path")
        start = int(x) if (x := m.group("start")) != "" else None
        end = int(x) if (x := m.group("end")) != "" else None
        return path, slice(start, end)
```

```
    return s, None
```

tests/test_read_file_slicing.py

新增 82 行单元测试，覆盖解析、非负切片、负切片及负边界大于文件等场景，是本次修复的回归保护。

```
# tests/test_read_file_slicing.py — 数据集切片语法回归测试（新增）

"""CPU 单元测试：path@[start:end] 数据集切片语法。

_parse_generalized_path 的正则显式接受两个边界的符号（-?\d*），且该语法
最初以 df.iloc[row_slice] 实现支持负索引；流式重写换成 itertools.islice 后
任何负索引都会抛 ValueError，导致 @[-100:]（取最后 100 行）从可用变为崩溃。
本测试固定以下行为：非负切片继续走 islice 流式路径；负边界切片按真实行数解析。
"""

from __future__ import annotations

import json

import pytest

from slime.utils.data import _parse_generalized_path, read_file


NUM_GPUS = 0 # CI 中标记为 CPU 测试

ROWS = [{"id": i} for i in range(10)]


@pytest.fixture
def jsonl_path(tmp_path):
    """构造一个 10 行的临时 jsonl 文件。"""
    path = tmp_path / "data.jsonl"
    path.write_text("".join(json.dumps(row) + "\n" for row in ROWS))
    return str(path)


def _ids(generalized_path):
    """辅助函数：读取文件并返回所有行的 id 列表。"""
    return [row["id"] for row in read_file(generalized_path)]


@pytest.mark.unit
def test_parse_generalized_path():
    """验证解析器对负边界的语法识别。"""
    assert _parse_generalized_path("/a/b.jsonl") == ("/a/b.jsonl", None)
    assert _parse_generalized_path("/a/b.jsonl@[3:7]") == ("/a/b.jsonl", slice(3, 7))
    assert _parse_generalized_path("/a/b.jsonl@[-100:]") == ("/a/b.jsonl", slice(-100, None))
```

```
    assert _parse_generalized_path("/a/b.jsonl@[:-2]") == ("/a/b.jsonl", slice(None, -2))


@pytest.mark.unit
def test_no_slice_reads_everything(jsonl_path):
    """不带切片时读取全部行。"""
    assert _ids(jsonl_path) == list(range(10))


@pytest.mark.unit
@pytest.mark.parametrize(
    "suffix,expected",
    [
        ("@[0:3]", [0, 1, 2]),
        ("@[3:]", [3, 4, 5, 6, 7, 8, 9]),
        ("@[:4]", [0, 1, 2, 3]),
    ],
)
def test_non_negative_slices(jsonl_path, suffix, expected):
    """非负切片保持流式 islice 路径，行为不变。"""
    assert _ids(jsonl_path + suffix) == expected


@pytest.mark.unit
@pytest.mark.parametrize(
    "suffix,expected",
    [
        ("@[-3:]", [7, 8, 9]),
        ("@[:-2]", [0, 1, 2, 3, 4, 5, 6, 7]),
        ("@[1:-1]", [1, 2, 3, 4, 5, 6, 7, 8]),
        ("@[-5:-2]", [5, 6, 7]),
    ],
)
def test_negative_slices(jsonl_path, suffix, expected):
    """负边界切片按 Python 列表语义解析。"""
    assert _ids(jsonl_path + suffix) == expected


@pytest.mark.unit
def test_negative_slice_larger_than_file(jsonl_path):
    """负边界超过文件行数时返回整个文件，与列表切片语义一致。"""
    assert _ids(jsonl_path + "@[-100:]") == list(range(10))
```

评论区精华

该 PR 没有 review 评论或讨论线程。PR body 本身是主要的设计说明，关键决策点是：负切片场景下物化 reader 以获取总行数，这一设计被作者论证为可行，因为 #696 之前的 pandas 实现本来就会为每个文件物化数据，因此对于受影响场景没有新增内存压力，而常见非负场景

仍保持流式处理。

- 暂无高价值评论线程

风险与影响

- 风险：技术风险主要体现在：
  1. 内存风险（限定场景）：负边界切片会物化整个 reader（list(reader)）。对于超大数据集（例如数 GB 的 jsonl），仅取末尾少量行也会全量载入内存，可能引发 OOM。虽然作者指出旧 pandas 实现同样物化，但流式重写的初衷正是避免物化，此修复在负边界场景后退了一步。
  2. 行为一致性：物化后使用 Python 列表切片，其语义与 itertools.islice 不完全一致（如 step 为负或反向切片）。当前 read_file 执行路径主要面向按行顺序生成数据，反向切片可能产生意外结果，但 parser 限制 start/end 均为非负或负整数且未显式禁止反向顺序。
  3. 对 parquet 路径的影响：该修改同样作用于 .parquet 文件的切片逻辑。pf.iter_batches() 物化后行序可能受批次影响，但列表切片语义仍成立。
  4. 测试覆盖局限：新测试仅覆盖 jsonl 文件，未覆盖 parquet 分支及 step 参数为负或其他边界组合。
- 影响：影响范围：
  1. 用户影响：所有使用 path@[start:end] 语法并依赖负索引（如取最后 N 行）的用户恢复可用，无需修改命令。
  2. 系统影响：read_file 是数据加载核心函数，被 --prompt-data、--eval-data 等参数广泛引用。非负切片路径完全不变，负切片路径的行为与 #696 之前的 pandas 语义一致，回归风险较低。
  3. 团队影响：新增测试明确了该语法的时间线（#526 引入、#696 流式化、本次修复恢复），为后续维护提供了回归保护。 - 风险标记：负切片物化可能引发内存风险，缺少 parquet 分支测试，测试只覆盖 jsonl 路径

关联脉络

- PR #2237 fix: keep dataset order in filter_long_prompt for mixed multimodal data: 同仓库近期 PR，同样修改了 slime/utils/data.py，涉及数据读取与处理的顺序 / 边界问题，可视为同一模块的稳定性演进。