

PR #2330 完整报告

THUDM/slime

[docs] fix out-dated doc

合并时间: 2026-08-26 15:23

原文链接: <http://prhub.com.cn/THUDM/slime/pull/2330>

执行摘要

- 一句话: 修复过期文档并新增文档一致性测试
- 推荐动作: 值得精读测试文件 `tests/test_docs_consistency.py` 的设计思路, 它用不到 100 行代码自动守护了全仓库文档的链接和命令有效性, 可复用到其他仓库; `docs/conf.py` 的 `_sync_examples` 变更也值得关注, 它展示了如何让示例文档引用真实文件而非仅有 README。若团队有类似文档漂移痛点, 可将此 PR 作为模板参考。

功能与动机

PR body 为空, 但测试文件 `docstring` 明确表达目标: **Guard documentation references that can be checked without a GPU runtime**. 从变更内容看, 文档已与当前实现脱节: 示例页链接指向带编号的旧锚点、`--prompt-data` 的描述未提及新增的 Parquet 格式、版本号停留在 0.0.1, 而实际发布版本已迭代到 0.3.1。此 PR 旨在修正这些漂移, 并建立自动化防线防止再次过期。

实现拆解

1. 新增文档一致性测试 (`tests/test_docs_consistency.py`): 新增 90 行测试文件, 包含三个核心测试——`test_local_markdown_links_exist` 校验所有本地 Markdown 链接指向存在的文件; `test_documented_script_and_test_commands_exist` 校验文档中出现的脚本 / 测试命令路径真实存在; `test_customization_anchor_links_exist` 校验 `customization.md` 中的锚点与标题匹配。测试通过 `NUM_GPUS = 0` 标记为无需 GPU 的只读检查。
2. 修正文档构建配置 (`docs/conf.py`): 将 `sys.path` 注入方式从 `os.path.abspath("../..")` 改为基于 `__file__` 的绝对路径, 提升可移植性; 版本号从 0.0.1 更新到 0.3.1; `_sync_examples` 中由 `mkdir` 改为 `shutil.copytree` 递归复制整个示例目录 (忽略两侧 README 文件), 使 `_examples_synced` 目录能承载示例中的辅助文件, 之后再覆盖语言特定的 README。
3. 更新参数帮助文本 (`slime/utils/arguments.py`): 将 `--prompt-data` 的帮助信息从「仅支持 jsonl」改为「支持 JSONL 和 Parquet (Parquet 需 pyarrow)」, 与实现保持同步。
4. 修复 `customization` 文档锚点 (`docs/en/get_started/customization.md`、`docs/zh/get_started/customization.md`): 将表格中的编号式锚点 (如 `#1-rollout-function---rollout-function-path`) 改为与标题自动生成的 slug 锚点 (如 `#rollout-function-path`), 避免锚点失效。

5. 接入 CI (.github/workflows/pr-test.yml、.github/workflows/pr-test.yml.j2)：在零 GPU 测试矩阵中新增 test_docs_consistency.py 条目，使文档一致性检查进入主流水线。

关键文件：

- tests/test_docs_consistency.py (模块 文档校验；类别 test；类型 test-coverage；符号 _markdown_files, _local_link_target, test_local_markdown_links_exist, test_documented_script_and_test_commands_exist)：新增的文档一致性测试，是本 PR 的核心防回归机制，覆盖链接、命令路径和锚点三类检查，并以零 GPU 方式接入 CI。
- docs/conf.py (模块 文档构建；类别 source；类型 core-logic；符号 _sync_examples)：更新文档构建配置：版本号同步到 0.3.1、路径处理更稳健，并将示例同步从仅复制 README 改为递归复制整个示例目录，影响文档站点结构。
- slime/utils/arguments.py (模块 参数解析；类别 source；类型 core-logic；符号 add_data_arguments)：更新 --prompt-data 参数的帮助文本，补充 Parquet 格式支持说明，使参数文档与实际能力一致。
- .github/workflows/pr-test.yml (模块 CI 流程；类别 infra；类型 infrastructure)：在零 GPU 测试矩阵中新增 test_docs_consistency.py，使文档一致性检查进入主 CI 流程，防止回归。
- docs/en/get_started/customization.md (模块 定制文档；类别 docs；类型 documentation)：修正英文版 customization 文档中的锚点，从带编号的旧锚点改为自动生成的 slug 锚点，并调整表格结构与内容。
- docs/zh/get_started/customization.md (模块 定制文档；类别 docs；类型 documentation)：与英文版同步修正中文 customization 文档锚点，保持两种语言一致性。

关键符号：_markdown_files, _local_link_target, test_local_markdown_links_exist, test_documented_script_and_test_commands_exist, test_customization_anchor_links_exist, _sync_examples

关键源码片段

tests/test_docs_consistency.py

新增的文档一致性测试，是本 PR 的核心防回归机制，覆盖链接、命令路径和锚点三类检查，并以零 GPU 方式接入 CI。

```
"""文档一致性校验测试：无需 GPU 即可运行的只读检查。"""
```

```
import re
from pathlib import Path
from urllib.parse import unquote
```

```
ROOT = Path(__file__).resolve().parents[1]
# 需要检查的文档来源：根目录 README、中英文 docs、examples 和 docker
DOC_SOURCES = (
    ROOT / "README.md",
    ROOT / "README_zh.md",
```

```

    ROOT / "docs" / "en",
    ROOT / "docs" / "zh",
    ROOT / "examples",
    ROOT / "docker",
)
# 识别 Markdown 链接 : ![alt](target)
MARKDOWN_LINK_RE = re.compile(r"!?\[[^\]]*\]\s*(([^\n]+\n)+)\s")
# 识别文档中出现的 bash/python 命令路径
COMMAND_PATH_RE = re.compile(
    r"\b(?:bash|python)\s+((?:scripts?|tests)/[A-Za-z0-9_+/-]+\.(?:sh|py))"
)

def _markdown_files():
    """按顺序产出所有需要检查的 Markdown 文件，跳过示例生成的同步目录。"""
    for source in DOC_SOURCES:
        if source.is_file():
            yield source
        else:
            # 对目录递归查找 *.md，同时排除 docs 构建时自动生成的 _examples_synced
            for path in sorted(source.rglob("*.md")):
                if "_examples_synced" not in path.parts:
                    yield path

def _local_link_target(raw_target: str) -> str | None:
    """从原始链接文本中提取本地文件路径。

    过滤掉锚点链接、绝对路径、URL、邮箱和 data URI；
    返回解码后的相对路径，便于与文件系统比对。
    """

    target = raw_target.strip()
    if target.startswith("<") and ">" in target:
        target = target[1:target.index(">")]
    else:
        target = target.split(maxsplit=1)[0] # 去掉标题和空白

    if target.startswith(("#", "/")) or "://" in target or target.startswith(("mailto:", "data:")):
        return None

    # 去掉 #anchor 和 ?query 后做 percent-decode
    return unquote(target.split("#", 1)[0].split("?", 1)[0]) or None

def test_local_markdown_links_exist():
    """校验文档中的所有本地相对链接都指向存在的文件。"""
    missing = []
    for markdown_file in _markdown_files():

```

```

text = markdown_file.read_text(encoding="utf-8")
for raw_target in MARKDOWN_LINK_RE.findall(text):
    target = _local_link_target(raw_target)
    if (
        target is not None
        and "_examples_synced" not in Path(target).parts
        and not (markdown_file.parent / target).exists()
    ):
        missing.append(f"{markdown_file.relative_to(ROOT)} -> {target}")
assert not missing, "Broken local Markdown links:\n" + "\n".join(missing)

```

```

def test_documented_script_and_test_commands_exist():
    """校验文档中出现的脚本/测试命令路径真实存在。"""
    missing = []
    for markdown_file in _markdown_files():
        text = markdown_file.read_text(encoding="utf-8")
        for command_path in COMMAND_PATH_RE.findall(text):
            if not (ROOT / command_path).is_file():
                missing.append(f"{markdown_file.relative_to(ROOT)} -> {command_path}")
    assert not missing, "Documented command paths do not exist:\n" + "\n".join(missing)

```

docs/conf.py

更新文档构建配置：版本号同步到 0.3.1、路径处理更稳健，并将示例同步从仅复制 README 改为递归复制整个示例目录，影响文档站点结构。

```

def _sync_examples(app):
    """在 Sphinx 构建前，将 examples 下的示例同步到各语言文档目录。

```

每个子目录优先使用语言对应的阅读文件（zh 用 README_zh.md），找不到时回退到 README.md；同时递归复制整个示例目录，以便文档可以引用示例中的其他辅助文件。

```

"""

```

```

if not src_dir.is_dir():
    return

```

```

for lang, cfg in lang_cfgs.items():
    lang_dir = cfg["dir"]
    if not lang_dir.exists():
        continue
    out_dir = lang_dir / "_examples_synced"
    # 每次构建前先清除旧目录，避免残留文件
    if out_dir.exists():
        shutil.rmtree(out_dir)
    out_dir.mkdir(parents=True, exist_ok=True)

```

```

for d in sorted(src_dir.iterdir()):
    if not d.is_dir():

```

```

        continue
    if lang == "zh":
        # 中文文档优先使用 README_zh.md, 缺失时回退 README.md
        primary = d / "README_zh.md"
        fallback = d / "README.md"
        candidate = primary if primary.exists() else fallback
    else:
        candidate = d / "README.md"
    if not candidate.exists():
        continue

    target_dir = out_dir / d.name
    # 递归复制示例目录, 排除两侧的 README, 随后统一覆盖为目标阅读文件
    shutil.copytree(d, target_dir, ignore=shutil.ignore_patterns("README.md", "README_
    zh.md"))
    shutil.copy2(candidate, target_dir / "README.md")
    entries.append((d.name, f"_examples_synced/{d.name}/README.md"))

```

评论区精华

该 PR 没有任何 review 评论或 thread, 作者直接合入。从最终状态可推断改动被维护者接受, 无需额外讨论。

- 暂无高价值评论线程

风险与影响

- 风险:

1. 文档构建逻辑变更: docs/conf.py 中 shutil.copytree 会递归复制整个示例目录, 若示例中包含大文件或非常规文件 (如隐藏文件、符号链接), 可能拖慢文档构建甚至触发构建错误; 同时 _examples_synced 目录内容增多, 可能引入意外文件暴露到文档站点。
2. 新增 CI 测试的误报风险: test_local_markdown_links_exist 使用正则解析 Markdown 链接, 遇到包含空格或特殊字符的合法链接可能误判为损坏; test_documented_script_and_test_commands_exist 严格校验命令行路径, 文档中如果演示「不存在的路径」作为示例会直接导致 CI 失败。
3. 参数帮助文本与实现的潜在不一致: slime/utils/arguments.py 仅更新了帮助文本, 若 Parquet 支持尚未合并或存在平台差异, 文档会形成新的误导。
4. 锚点变更的对外兼容性: customization.md 的锚点从带编号改为 slug 形式, 若外部链接或书签引用了旧锚点, 将变成死链。- 影响: 用户影响: 文档更贴近当前实现, 减少配置时的困惑, 尤其是 --prompt-data 的 Parquet 说明和示例命令的修正; 系统影响: CI 新增零 GPU 快速测试, 对整个流水线耗时影响极小; 团队影响: 文档维护成本降低, 死链和过期命令能在合入前被发现, 长期提升文档可信度。- 风险标记: 文档构建逻辑变更, 新增 CI 测试, 链接规则可能产生误报

关联脉络

- PR #2326 [doc] update doc: 同样属于文档维护类 PR, 更新 README 与 CONTRIBUTING, 与本 PR 共同组成近期文档改进线。
- PR #2297 fix: reject non-positive rollout temperature at parse time: 修改了 `slime/utils/arguments.py`, 与当前 PR 在同一文件有交集, 便于追踪参数相关变更的演化。