

PR #38824 完整报告

sgl-project/sglang

pyproject(xpu): drop human-eval git dep to unblock image build

合并时间: 2026-09-10 16:24

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/38824>

执行摘要

- 一句话: 移除 XPU pyproject 中的 human-eval git 依赖, 修复镜像构建失败
- 推荐动作: 值得快速浏览而非精读: 单个 TOML 文件、单行删除、零逻辑变更, 阅读成本极低, 重点是理解失败根因 (setuptools >= 81 移除 pkg_resources 与 PEP 517 隔离构建的交互), 这个模式对任何依赖老式 setup.py + git 直装的仓库都有复用价值。阅读时建议关注两点: (1) 是否应在 test extra 处留下注释, 标注 human-eval 被临时摘除及其恢复条件; (2) 是否存在替代路径 (如换成带 pyproject.toml 的 fork、或在 Dockerfile 里固定 setuptools < 81) 比直接删除依赖更完整。若你在 XPU 上依赖 human-eval 做评测, 需自行补装并留意运行期 ImportError。

功能与动机

PR body 明确说明: XPU nightly Docker 镜像构建在 docker/xpu.Dockerfile 第 78-85 行的 `pip install .[dev,diffusion]` 步骤失败, 报错 `ModuleNotFoundError: No module named 'pkg_resources'`, 失败运行链接为 actions/runs/34447556590。根因是 merrymercy/human-eval 未提供 pyproject.toml, 其 setup.py 在模块加载时 `import pkg_resources`, 而 setuptools >= 81 把 pkg_resources 移出默认安装, pip 的 PEP 517 隔离构建环境会拉到更新版 setuptools, 于是在读取 requirements.txt 之前就崩溃。同一故障也会命中任何针对 pyproject_xpu.toml 的本地 `pip install .[test]`。目标是先解除镜像构建阻塞, 待上游补齐 pyproject.toml 或替换为现代打包分支后再引入。

实现拆解

1. 唯一变更入口: python/pyproject_xpu.toml 的 [project.optional-dependencies] 中 test 数组删除一项 git 直装依赖 human-eval @ git+https://github.com/merrymercy/human-eval.git, 这是本 PR 全部代码改动 (+0/-1), 未触碰任何 Python 源码。
2. 依赖传播链被一并切断: 该文件随后声明 `dev = ["sglang[test]"]`、`all = ["sglang[diffusion]", "sglang[tracing]"]`, 因此 test extra 的收缩会自动传导到 dev, 使 docker/xpu.Dockerfile 中的 `pip install .[dev,diffusion]` 不再解析该 git 依赖, 从而绕过 PEP 517 隔离构建。此处的关键设计是「改叶子 extra、不动 Dockerfile」, 把修复点收敛在依赖声明层, 避免镜像脚本分叉。
3. 未做配套改动: 没有新增测试、没有改 Dockerfile、没有修订文档, 也没有在 test extra 中添加 human-eval 的替代包。PR body 中「Add unit tests」「Update documentation」「Provide accuracy and speed benchmark results」三项 checklist 均未勾选, 属于典型

的构建阻塞型最小修复。

4. 对下游的隐性契约变化：任何依赖 `pyproject_xpu.toml` 的 `test/dev` extra 来跑评测的 XPU 流程，将不再具备 `human-eval` 可用性；若 CI 或脚本中存在 `import human_eval` 的使用点，会从「pip 安装期失败」变为「运行期 ImportError」。本次 diff 中没有任何针对该风险的兜底或说明性注释。

关键文件：

- `python/pyproject_xpu.toml`（模块 依赖配置；类别 `config`；类型 `configuration`）：本 PR 唯一改动文件；从 `[project.optional-dependencies]` 的 `test` extra 中删除 `human-eval` git 直装依赖，直接决定 XPU 变体的依赖解析结果与 `docker/xpu.Dockerfile` 的安装成败。

关键符号：未识别

关键源码片段

`python/pyproject_xpu.toml`

本 PR 唯一改动文件；从 `[project.optional-dependencies]` 的 `test`extra 中删除 `human-eval` git 直装依赖，直接决定 XPU 变体的依赖解析结果与 `docker/xpu.Dockerfile` 的安装成败。

```
# python/pyproject_xpu.toml —— XPU 变体的可选依赖声明（整理后的关键片段）
[project.optional-dependencies]
diffusion = [
    "addict==2.4.0",
    "av==16.1.0",
    "diffusers==0.38.0",
    "PyYAML==6.0.1",
    "runai_model_streamer>=0.15.5",
    # st_attn 仅在非 ARM 平台安装，与主工程保持一致的平台标记写法
    "st_attn==0.0.7 ; platform_machine != 'aarch64' and platform_machine != 'arm64'",
    "trimesh>=4.0.0",
    "xatlas",
]

tracing = [
    "opentelemetry-api",
    "opentelemetry-exporter-otlp",
    "opentelemetry-sdk",
]

test = [
    "accelerate",
    "bitsandbytes",
    "pymupdf",
    "expecttest",
    # 本次 PR 删除的条目（原位置在此处）：
    # "human-eval @ git+https://github.com/merrymercy/human-eval.git"
    # 删除原因：该上游仓库没有 pyproject.toml，只要走 git 直装，pip 就会为它
    # 创建 PEP 517 隔离构建环境并拉入新版 setuptools；而 setuptools >= 81 不再
```

```

# 默认提供 pkg_resources, 其 setup.py 在模块加载阶段的 `import pkg_resources`
# 会直接抛 ModuleNotFoundError, 使整个 `pip install .[dev,diffusion]` 失败,
# 进而阻塞 docker/xpu.Dockerfile 中 intel/sglang-dev 镜像的构建。
# 恢复条件: 上游补齐 pyproject.toml, 或替换为现代打包方式的 fork。
"jsonlines",
"lm-eval[api]>=0.4.9.2",
"matplotlib",
"pandas",
"parameterized",
"peft>=0.18.0",
"polars",
"pytest",
"sentence_transformers",
"tabulate",
]

# dev 通过 sglang[test] 间接继承 test 依赖, 因此上面的一次删除
# 同时让 `pip install .[dev]`、`.`[dev,diffusion]` 都不再拉取该 git 依赖。
dev = ["sglang[test]"]

# all 不包含 test/dev, 故本次改动不影响 all 的解析结果。
all = [
    "sglang[diffusion]",
    "sglang[tracing]",
]

```

评论区精华

本 PR 的 `comments_count`、`review_comments_count` 均为 0, Review 审核与 Review 评论数组为空, 没有任何人工评审交锋, 由 mingfeima 直接合并。唯一可提炼的「讨论性内容」来自 PR body 的根因陈述: 作者指出上游 `merrymercy/human-eval` 缺少 `pyproject.toml`、其 `setup.py` 在加载期 `import pkg_resources`, 而 `setuptools >= 81` 已移除默认的 `pkg_resources`, 故 `pip` 的 PEP 517 隔离环境必然失败。PR body 同时给出后续条件——「It can be reintroduced once merrymercy/human-eval grows a `pyproject.toml` (or is swapped for a modern packaging fork)」, 即这是一次有意识、可回滚的临时摘除, 而非永久放弃该依赖。

另一条值得记录的信号是 PR body 内嵌的 CI 状态: `Latest PR Test (Base)` 通过, 但 `Latest PR Test (Extra)` 与 `Latest PR Test (AMD ROCm 10)` 显示失败, PR 内未对本行改动是否与这两条失败相关做任何解释。

- 无人工 review, 仅以 PR body 陈述根因 (other): 作者给出的根因与修复方向 (先摘除 git 依赖解除阻塞) 自治, 由 mingfeima 直接合并, 无人提出异议, 也未留下替代方案 (如固定 `setuptools < 81`) 的讨论记录。
- CI 状态中 Extra 与 AMD ROCm 10 流水线失败未做说明 (testing): 未解决: 无法从现有材料判断这两条失败是仓库既有噪声还是依赖删除的副作用, 需依赖合并后的主干流水线验证。

风险与影响

- 风险:

1. 测试能力缺口（主要风险）：python/pyproject_xpu.toml 的 test extra 移除 human-eval 后，XPU 平台上任何基于该 extra 安装的评测流程将丢失 human_eval 模块。若 CI 中存在引用该模块的用例，故障形态会从构建期 ModuleNotFoundError: pkg_resources 迁移为运行期 ImportError: human_eval，即错误被推迟而非消除。
1. CI 失败状态未闭环：PR body 的 CI 面板显示 Extra 与 AMD ROCm 10 两条流水线为 :x:，而 PR 内既无 review 评论也无补充说明，无法确认这两条失败是仓库既有噪声还是本次依赖删除引入的副作用。合并后需靠主干流水线自证。
2. 依赖清单语义弱化：这是「以删除依赖换取构建成功」的权宜修复，未在 pyproject_xpu.toml 内留下注释标记该缺失项，后续维护者可能难以察觉 XPU 与主 pyproject.toml 的 test extra 已出现能力差异（需注意：本 PR 未提供主 pyproject.toml 的对照，差异范围存在不确定性）。
3. 回归风险低：改动不触及运行时代码、不改变 diffusion、tracing、all 等 extra 内容，也不影响 [tool.setuptools.package-data] 与 [project.scripts]，对推理路径零影响。- 影响：用户侧：XPU 用户执行 pip install .[test] 或使用 intel/sglang-dev:latest 镜像不再因 human-eval 中断安装，安装成功率提升；代价是 XPU 开发环境默认不含 human-eval，需要该包的用户得手动从 git 安装。系统侧：解除 XPU nightly Docker 镜像构建的硬阻塞，恢复 intel/sglang-dev 镜像的持续产出，间接影响依赖该镜像的所有 XPU CI 作业。团队侧：属于平台分支（XPU）依赖维护，影响范围限定在 Intel XPU 相关维护者；由于改动仅一行且已被 maintainer 直接合并，沟通成本极低，但缺少注释会让「临时摘除」的意图随时间流失。- 风险标记：测试依赖被移除，XPU 镜像构建阻塞修复，无 review 讨论，CI 失败未解释，缺注释易被回退

关联脉络

- PR #38796 docker(xpu): install libssl-dev so JIT hicache_hash_cpp builds: 同为 XPU 镜像构建阻塞类修复，改动 docker/xpu.Dockerfile 与 pr-test-xpu workflow；与本 PR 属于同一条「让 intel XPU 镜像能装起来」的维护线，可对照看 XPU 镜像依赖缺口的整体收敛情况。
- PR #38665 docker(xpu): drop redundant setvars.sh from torch_memory_saver RUN: 同样是为修复 XPU Dockerfile 中依赖安装失败而做的精简改动，思路一致：移除或修正阻塞安装的片段，而非重写镜像流程。
- PR #36098 Add missing test dependencies to pyproject.toml variants: 直接操作 pyproject 变体文件（python/pyproject_xpu.toml 同族）的依赖清单维护，本 PR 是该文件依赖集合的又一次调整，两者合起来反映 XPU 变体依赖清单长期落后于主 pyproject.toml 的问题。
- PR #38350 [HiCache] Fix side pools to use resolved host allocator: 关联度较弱但在同一批近期变更中：体现仓库当前对「配置 / 解析结果一致性」的持续关注，与本 PR 在配置层做最小定点修复的风格相近，可作为同一时期配置类修复的参照。