

PR #7650 完整报告

verl-project/verl

[ci] chore: upgrade transformers==5.9.0

合并时间: 2026-09-01 15:51

原文链接: <http://prhub.com.cn/verl-project/verl/pull/7650>

执行摘要

- 一句话: 统一项目所有后端的 transformers 版本至 5.9.0, 简化依赖管理。
- 推荐动作: 这是一个重要的依赖维护 PR, 虽然代码变更简单, 但影响面较广。建议精读 pyproject.toml 中关于 override-dependencies 新策略的注释, 理解其设计意图。对于所有使用者, 需要在更新环境后运行测试套件, 特别是与 transformers 交互紧密的功能, 以确保升级无破坏性。该 PR 是项目依赖治理简化的积极一步, 值得肯定。

功能与动机

动机是解决项目内不同后端 (vllm、sglang、cpu) 因 pin 不同 transformers 版本 (5.3.0, 5.5.3) 而带来的依赖管理复杂性和潜在冲突。通过升级到一个统一的、更新的版本 (5.9.0), 旨在简化环境配置并保持所有后端使用相同的基准库版本。

实现拆解

1. 核心配置更新: 在 pyproject.toml 中, 将 vllm、sglang、cpu 等 extra 中显式声明的 transformers 版本从旧值 (5.5.3, 5.3.0) 统一更新为 5.9.0。同时, 对 fsdp 和 megatron 等未显式声明 transformers 的 extra 的注释进行了相应更新, 明确说明版本由全局 override 决定。
2. 全局依赖覆盖策略变更: 这是最关键的变更。原先 override-dependencies 针对不同 extra 设置了不同的 transformers 覆盖规则 (如 extra == 'vllm' 为 5.5.3, extra == 'sglang' 为 5.3.0)。PR 将其简化为一条无条件规则 transformers==5.9.0, 这利用了 uv override 的绝对优先级, 强制项目内所有位置 (包括传递依赖) 的 transformers 版本都为 5.9.0。注释详细解释了此变更的原因和注意事项。
3. 同步更新关联依赖: 根据新版 transformers (5.9.0) 的兼容性, 将 kernels 的版本约束从 $\geq 0.10.2, < 0.11$ 调整为 $\geq 0.12.0, < 0.13$, 以避免已知的兼容性问题 (如 [huggingface/transformers#46291](https://github.com/huggingface/transformers/pull/46291))。
4. 重新生成锁文件: 执行 uv lock 命令, 基于更新后的 pyproject.toml 重新生成了 uv.lock 文件, 其大规模变更 (+1126/-1134) 反映了整个依赖树的重新解析和版本锁定。

关键文件:

- pyproject.toml (模块 依赖配置; 类别 config; 类型 configuration): 项目依赖配置的核心文件, 所有关于 transformers 版本的声明和全局覆盖策略变更都在此文件中, 是本次 PR 变更的源头和关键。

- uv.lock (模块 依赖锁; 类别 other; 类型 core-logic) : 由 uv 根据 pyproject.toml 变更自动生成的锁文件, 其大规模变更反映了依赖解析结果, 是验证依赖升级是否正确的最终产物。

关键符号: 未识别

关键源码片段

pyproject.toml

项目依赖配置的核心文件, 所有关于 transformers 版本的声明和全局覆盖策略变更都在此文件中, 是本次 PR 变更的源头和关键。

```
# vllm extra 变更: 从 5.5.3 升级至 5.9.0, 并更新注释说明这是项目全局 pin。
```

```
vllm = [  
    # ... 其他依赖  
    # vllm 0.24 needs transformers>=5.5.3; 5.9.0 is the project-wide pin  
    # (enforced in override-dependencies below).  
    "transformers==5.9.0",  
]
```

```
# sglang extra 变更: 从 5.3.0 升级至 5.9.0, 注释说明覆盖了 sglang 自身的 pin。
```

```
sglang = [  
    # ... 其他依赖  
    # sglang 0.5.12 hard-pins transformers==5.6.0; the project-wide override below  
    # replaces that pin with 5.9.0 (5.6.0 is excluded repo-wide, see setup.py).  
    "transformers==5.9.0",  
]
```

```
# cpu extra 变更: 从 5.3.0 升级至 5.9.0。
```

```
cpu = [  
    # ... 其他依赖  
    "transformers==5.9.0",  
]
```

```
# 全局依赖覆盖策略的核心变更:
```

```
# 旧策略: 针对不同 extra 指定不同版本 (5.3.0, 5.5.3)
```

```
# 新策略: 一条规则强制所有位置为 5.9.0, 利用 override 的绝对优先级。
```

```
override-dependencies = [  
    # * transformers==5.9.0 project-wide — one version for every backend, so a  
    # venv's transformers no longer depends on which engine it was synced with.  
    "transformers==5.9.0",  
    "numpy>=2.0.0",  
    # * kernels 版本约束同步更新以匹配 transformers 5.9.0 的兼容性要求。  
    "kernels>=0.12.0,<0.13",  
    # ... 其他 override  
]
```

评论区精华

该 PR 没有留下任何 review 讨论或评论。所有决策似乎在提交前已完成。

- 暂无高价值评论线程

风险与影响

- 风险：兼容性风险（中等）：transformers 从 5.3.x/5.5.x 升级到 5.9.0 跨了多个版本，尽管注释表明该版本仍在 `setup.py` 定义的支持窗口内 ($\geq 5.5.3, \neq 5.6.0, < 5.11$)，但仍可能存在 API 弃用、行为变化或与现有代码（尤其是 `verl` 内部或第三方库如 `trl`、`sglang`、`vllm`）的未预料的交互。特别是 `sglang 0.5.12` 原先硬 `pin transformers==5.6.0`，现在被全局 `override` 为 5.9.0，可能带来潜在的兼容性问题。依赖解析风险（低）：`uv.lock` 的大规模变更意味着所有相关包的版本树被重新解析。虽然工具（`uv`）旨在保证确定性，但如此大幅度的更新仍需在所有 CI 和本地环境中验证安装的稳定性和正确性。无运行时或安全风险：此变更仅涉及 Python 依赖声明和解析，不直接修改应用程序源代码逻辑，因此不引入新的运行时逻辑漏洞或安全问题。
- 影响：环境影响：所有开发者、CI/CD 流水线在拉取此变更后，都需要重新安装或同步环境（`uv sync`），以获取一致的 `transformers==5.9.0` 版本。系统影响：升级影响所有使用 `verl` 库的下游项目，它们将获得一个更新的 transformers 基线。这可能带来新特性或性能改进，但也可能要求下游项目检查其自身代码的兼容性。团队影响：极大地简化了 `verl` 项目的依赖管理。之前需要为不同后端（`vllm`、`sglang`、`cpu`）维护不同的 transformers 版本规则，现在统一为一条，降低了维护成本和出错概率。未来的版本升级只需修改一处。
- 风险标记：跨后端依赖升级，全局版本覆盖变更，第三方库兼容性待验证

关联脉络

- PR #7625 [model] fix: import AutoModelForCausalLMWithValueHead from `trl.experimental.ppo` in `utils/model.py`: 同样涉及 transformers 相关导入的修复，与依赖版本和 API 兼容性相关。
- PR #7627 [misc] feat: uv support aarch64: 与本 PR 同属项目依赖和构建基础设施的维护工作，涉及 `pyproject.toml` 和 `uv.lock` 的管理。
- PR #7629 [ci] chore: fix ci failure: 本 PR 的依赖升级可能影响 CI 流程，此 PR 修复了其他 CI 问题，两者在 CI 稳定性维护上有关联。
- PR #7584 [ci] fix: drop stale `enable_chunked_prefill=False` from Ascend NPU scripts: 同属清理过时配置和维护 CI 的范畴，与本 PR 的依赖清理性质相似。