

PR #48977 完整报告

vllm-project/vllm

[Mypy Fix] Mypy fix for "vllm/model_executor/models/[aA][bB]"

合并时间: 2026-08-07 03:30

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/48977>

执行摘要

- 一句话: 修复模型层 A/B 系列 mypy 错误, 收紧类型契约
- 推荐动作: 值得精读 interfaces.py 与 adapters.py 的改法: 如何用命名 Protocol 保留参数名、用 getattr 兜底避免类级可变默认值、拆分私有加载方法以消除 MRO 扫描 hack。维护者应关注 Bailing / AXK 系列 config 兼容性、supports_pp 判定回归, 以及外部对 ModelForPooling.load_weights 的覆写兼容性。后续同类 mypy PR 可参考本 PR 的 review 标准: 优先修复类型不一致本身, 而不是用 type: ignore 掩盖。

功能与动机

issue #26533 指出当前 tools/pre_commit/mypy.py 将若干目录放在 SEPARATE_GROUPS 中、import 不跟随, 导致本地单文件检查与全量检查不一致, 需要逐步把目录迁入 FILES 并修复暴露出的全部错误。本 PR 的 body 给出了修复前基线: bailing_moe_mtp.py 报 Unexpected keyword argument "shard_id" 和 str/int 赋值冲突, 共 115 errors in 19 files (checked 824 source files); 修复后 "Run mypy for Python 3.13... Passed"。该 PR 正是 issue 清单中未勾选的最后一环。

实现拆解

1. 接口层重构 (interfaces.py): 新增 MambaStateShapes TypeAlias 与 _MakeEmptyIntermediateTensors Protocol; SupportsPP 的 make_empty_intermediate_tensors 从方法改为协议属性, forward 返回类型扩展为 `Tensor | IntermediateTensors | tuple[Tensor, list[Tensor]]`; _supports_pp_attributes 增加 `SupportsPP in model.__mro__` 判断; SupportsQuant.packed_modules_mapping 去掉默认值并在 _maybe_apply_model_mapping 中用 getattr 兜底。这些改动让 PP 与量化接口在类型系统下更精确, 同时保留参数名可读性。
2. 适配器拆分 (adapters.py): ModelForPooling.load_weights 拆分为 _load_pooling_model_weights (承担原有加载逻辑) 和新 load_weights (转发入口); load_weights_using_from_2_way_softmax 与 load_weights_no_post_processing 删除“按 `__name__ == "ModelForPooling"` 扫描 MRO”的 hack, 直接调用 `model._load_pooling_model_weights(weights)`; _create_pooling_model_cls / as_embedding_model / as_seq_cls_model 签名从 `_T` 泛型收紧为 `type[_T]`, 并用 `multimodal_model: object` 中转多模态检测。

3. 模型文件类型收窄：Bailing MoE 系列（bailing_moe.py / bailing_moe_linear.py / bailing_moe_v3.py / bailing_moe_mtp.py / bailing_moe_v3_mtp.py）为 fused_qkv_a_proj、q_a_layernorm、shared_experts 等条件创建成员标注 `X | None`；weight_loader 统一注解为 `Callable[..., None]`；将循环内的 shard_id 重命名为 stacked_shard_id / expert_shard_id 避免类型冲突；bailing_moe_linear.py 在 score_function 为 None 时补默认值 "softmax"，bailing_moe_v3_mtp.py 的 forward 将 hidden_states 改为可选并用 assert 收窄。
4. 配置字段断言（AXK1.py 等）：对 n_routed_experts、rope_parameters、max_position_embeddings 等配置字段增加 `assert ... is not None`，把隐性运行期错误提前到初始化阶段并满足类型收窄；_is_layer_sparse、load_weights 映射等位置同步补断言与变量改名。
5. 多模态与辅助模型：audioflamingo3.py / blip2.py 把 `audio_input["type"]` 等字典索引改为属性访问（.type / .data / .audio_embeds），并对 dummy options 做 isinstance 收窄；bert_with_rope.py 将 jina_merge_lora_weights 内的局部变量 weights 重命名为 weights_dict 避免与迭代器类型冲突；aria / idfics2_vision_model / llama / bagel / bee / aimv2 / arctic / afmoe / llama_next 等为小改动（导入调整、类型断言）。测试配套：本 PR 未新增测试文件，依靠 mypy 3.13 全量检查与 Buildkite 模型级 CI 回归（24 个 commit 中多次 merge main 与 retry CI 后合并）。

关键文件：

- vllm/model_executor/models/adapters.py（模块 模型层；类别 source；类型 data-contract；符号 _create_pooling_model_cls, ModelForPooling, load_weights, _load_pooling_model_weights）：将 ModelForPooling.load_weights 拆分为 _load_pooling_model_weights 并新增转发入口，删除 seq_cls 加载路径中的 MRO 扫描 hack，是 embedding/seq_cls 适配的公共入口变更。
- vllm/model_executor/models/interfaces.py（模块 模型层；类别 source；类型 data-contract；符号 _MakeEmptyIntermediateTensors, call, make_empty_intermediate_tensors）：公共接口契约调整：SupportsPP 协议、SupportsQuant 的 packed_modules_mapping 默认值处理、_supports_pp_attributes 的 MRO 判断，影响所有支持 PP/ 量化的模型。
- vllm/model_executor/models/bailing_moe_linear.py（模块 模型层；类别 source；类型 data-contract）：为条件创建成员补充可选类型标注，score_function 新增默认值，是 Bailing MoE v2.5 的核心实现。
- vllm/model_executor/models/bailing_moe_v3.py（模块 模型层；类别 source；类型 data-contract）：KDA 注意力与 Mamba 状态形状类型修正，新增多处 assert 收窄，是 Bailing MoE v3 的核心路径。
- vllm/model_executor/models/AXK1.py（模块 模型层；类别 source；类型 data-contract）：大量 config 字段断言与 Optional 标注，影响 AXK1 模型初始化与 load_weights 映射。
- vllm/model_executor/models/bailing_moe_v3_mtp.py（模块 模型层；类别 source；类型 data-contract）：forward 签名将 hidden_states 改为可选并加 assert，MTP 权重加载路径变量改名，属于投机解码配套。

- `vllm/model_executor/models/bailing_moe_mtp.py` (模块 模型层; 类别 source; 类型 data-contract) : 修复 PR body 中提到的 Unexpected keyword argument "shard_id" 与 str/int 冲突问题。
- `vllm/model_executor/models/audioflamingo3.py` (模块 模型层; 类别 source; 类型 data-contract) : 多模态处理器从 dict 索引改为属性访问并补充类型收窄, 是 AudioFlamingo3 输入处理路径的行为修正。

关键符号: `_create_pooling_model_cls`, `_load_pooling_model_weights`, `load_weights`, `as_embedding_model`, `as_seq_cls_model`, `_get_language_model_for_seq_cls`, `supports_pp`, `_supports_pp_attributes`, `_maybe_apply_model_mapping`, `_MakeEmptyIntermediateTensors`

关键源码片段

`vllm/model_executor/models/interfaces.py`

公共接口契约调整: SupportsPP 协议、SupportsQuant 的 `packed_modules_mapping` 默认值处理、`_supports_pp_attributes` 的 MRO 判断, 影响所有支持 PP/ 量化的模型。

`vllm/model_executor/models/interfaces.py` (合并后整理)

```
class _MakeEmptyIntermediateTensors(Protocol):
    # 显式协议替代裸 Callable: 保留 batch_size / dtype / device 参数名,
    # 满足“类型信息不丢失”的 review 要求, 同时解决 method-assign 报错。
    def __call__(
        self,
        batch_size: int,
        dtype: torch.dtype,
        device: torch.device,
    ) -> "IntermediateTensors": ...
```

@runtime_checkable

```
class SupportsPP(Protocol):
    """支持 pipeline parallel 的模型统一接口。"""

    supports_pp: ClassVar[Literal[True]] = True
    make_empty_intermediate_tensors: _MakeEmptyIntermediateTensors
    """Called when PP rank > 0 for profiling purposes."""

    def forward(
        self,
        input_ids: Tensor | None,
        positions: Tensor,
        *,
        intermediate_tensors: "IntermediateTensors | None",
    ) -> "Tensor | IntermediateTensors | tuple[Tensor, list[Tensor]]":
        """PP rank > 0 时接收中间张量, 最后 rank 返回输出。"""
        ...
```

```
def _supports_pp_attributes(model: type[object] | object) -> bool:
    if isinstance(model, type):
        # 除协议检查外增加 MRO 判断: 直接继承 SupportsPP 的类
        # 即使未显式声明 make_empty_intermediate_tensors 属性也视为支持 PP
        return SupportsPP in model.__mro__ or isinstance(model, _SupportsPPType)
    return isinstance(model, SupportsPP)

# SupportsQuant._maybe_apply_model_mapping 中的配套改动:
# packed_modules_mapping 不再给类级默认 {} (避免可变默认值共享),
# 用 getattr 兜底, 缺失时直接跳过更新
if packed_modules_mapping := getattr(self, "packed_modules_mapping", None):
    self.quant_config.packed_modules_mapping.update(packed_modules_mapping)
```

评论区精华

- hmellor 在整体 review 中要求减少 type: ignore: "IMO there are a too many type: ignores, we should really be trying to fix whatever the inconsistency is, that is the purpose of this effort". 作者随后提交 "remove type ignore", 将多数 ignore 替换为显式类型收窄, 仅保留动态子类等少量必要注释。
- hmellor 指出 packed_modules_mapping: ClassVar[dict[str, list[str]]] = {} 是可变默认值: "Mutable default is not safe". 作者改为不设默认值, 并在 _maybe_apply_model_mapping 中用 getattr(self, "packed_modules_mapping", None) 兜底。
- hmellor 认为把 make_empty_intermediate_tensors 改成裸 Callable 丢失参数名: "What was the error here? I think retaining the argument names is important so this change causes us to lose information". 作者引入 _MakeEmptyIntermediateTensors Protocol, 保留 batch_size / dtype / device 参数名并解决 method-assign 错误。
- hmellor 建议用 TypeIs 收窄 supports_multimodal, 作者解释会报 Missing positional argument "self", hmellor 接受: "Hmm, I guess mypy is not being smart enough then", 最终保留 object 中转方案。
- hmellor 对多处 weight_loader: Any 追问 "Callable?", 作者统一改为 Callable[..., None]。
- hmellor 询问 Mamba 状态形状是否确实变化 ("Do these shapes vary?"), 作者用 MambaStateShapes 联合别名表达合法形状集合, 避免过度泛化。
- type: ignore 数量过多 (design): 作者新增 commit "remove type ignore", 将多数 ignore 替换为显式类型收窄, 仅保留动态子类等少数必要注释。
- 类级可变默认值 (correctness): 作者改为不设默认值, 并在 _maybe_apply_model_mapping 中用 getattr(self, "packed_modules_mapping", None) 兜底。
- make_empty_intermediate_tensors 签名信息保留 (design): 作者新增 _MakeEmptyIntermediateTensors Protocol, 保留 batch_size / dtype / device 参数名并解决 method-assign。

- supports_multimodal TypeIs 收窄 (design): hmellor 认可 mypy 局限 ("mypy is not being smart enough then") , 保留 multimodal_model: object 中转方案。
- weight_loader 注解类型 (style): 作者统一改为 Callable[..., None], 保留 default_weight_loader 兜底。
- Mamba 状态形状是否变化 (correctness): 作者用 MambaStateShapes 联合别名表达合法形状集合, 避免单一 Tuple[...] 泛化。

风险与影响

- 风险:
 1. 运行时行为微调: bailing_moe_linear.py 在 score_function 为 None 时新增默认值 "softmax"; bailing_moe_v3_mtp.py 的 forward 签名变为 Optional + assert; AXK1.py 与 bailing_moe_v3.py 新增多条 assert ... is not None, 可能使非标准 checkpoint 从启动期静默错配变成初始化即抛 AssertionError。
 2. PP 检测语义变化: interfaces.py 的 _supports_pp_attributes 增加 SupportsPP in model.__mro__ 判断, 会让少量未显式声明 make_empty_intermediate_tensors 但继承 SupportsPP 的类被识别为支持 PP, 需观察 pipeline parallel 路径是否出现行为回归。
 3. 适配器入口变化: adapters.py 将 load_weights 拆分为转发结构, 外部自定义模型若覆写 load_weights 并调用 super().load_weights(), 现在会走 _load_pooling_model_weights; 若外部代码曾依赖 MRO 扫描 dynamic 类, 存在兼容隐患 (PR 内引用已同步更新)。
 4. 多模态输入访问方式变更: audioflamingo3.py / blip2.py 从 dict 索引改为属性访问 (.type / .data / .audio_embeds), 若 AudioFlamingo3Inputs / Blip2ImageInputs 在个别路径仍以 dict 传入将抛 AttributeError; 从类型定义看应为 dataclass, 风险较低。
 5. 缺少直接测试配套: 23 个文件零测试文件变更, 回归主要依赖 mypy 3.13 全量检查与 Buildkite 模型级 CI。- 影响: 对最终用户: 推理行为基本不变, 仅异常路径的报错时机和个别默认值有差异。对系统开发者: mypy 3.13 全量检查在本目录通过是本系列的质量里程碑, 后续贡献者在本地编辑这些模型文件时能获得更准确的类型检查; adapters.py 与 interfaces.py 的契约变化 (load_weights 拆分、SupportsPP 协议调整) 会被后续模型与 embedding/seq-cls 相关 PR 依赖。对团队: 该 PR 改动面广 (23 文件、24 commits), 过程中多次 merge main 与 CI retry, 与 main 同步成本较高, 最终由作者自行合并。- 风险标记: 公共协议变更, 运行时语义微调, 无新增测试配套, 多模态输入访问方式变更

关联脉络

- 暂无明显关联 PR