PR #24926 完整报告

sgl-project/sglang

spec: centralize EagleDraft{,Extend}Input.hidden_states shape

合并时间：2026-05-11 13:49

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24926

执行摘要

- 一句话：集中 EagleDraftInput hidden_states 形状决策
- 推荐动作：建议精读该 PR，了解 SGLang speculative decoding 中 hidden_states 的形状决定逻辑，以及如何通过类方法实现单一真相来源的设计模式。同时为理解后续 PR（#21058 相关）提供基础。

功能与动机

PR 描述明确指出需要单一真相来源（single source of truth），消除分散的重复形状计算，并为后续 PR（关联 #21058）中让 hidden_states 变为 Optional 做准备。作者在 Roadmap 中规划了三个 PR，本 PR 为第一步。

实现拆解

1. 添加统一访问辅助函数（eagle_info.py）：引入 _draft_runner_of(worker)，处理 v1（EAGLEWorker 直接暴露 model_runner）与 v2（EagleDraftWorker 暴露 draft_runner）命名差异。
2. 新增类方法集中形状决策（eagle_info.py）：在 EagleDraftInput 和 EagleDraftExtendInput 中分别添加 hidden_size_for 和 dtype_for 类方法。decode 阶段返回 draft 模型的 spec_hidden_size；extend 阶段根据算法返回 target 模型的 spec_hidden_size，EAGLE-3 aux 模式返回 target.hidden_size * 3。
3. 替换调用点（eagle_worker.py）：_draft_preprocess_idle 和 forward_draft_extend_after_decode 中原先直接访问 self.model_config 的逻辑改为调用 EagleDraftInput.hidden_size_for(self) / EagleDraftExtendInput.hidden_size_for(self)。
4. 更新 CUDA 图静态 buffer 初始化（eagle_draft_cuda_graph_runner.py 和 eagle_draft_extend_cuda_graph_runner.py）：原本分散的条件分支（如 EAGLE-3 aux 判断）被替换为统一的类方法调用。

无测试文件改动，CI 已通过 speculative decoding 相关测试。

关键文件：

- python/sglang/srt/speculative/eagle_info.py（模块 推测解码；类别 source；类型 core-logic；符号 _draft_runner_of, hidden_size_for, dtype_for）：核心文件：新增 _draft_runner_of 辅助函数以及 EagleDraftInput 和 EagleDraftExtendInput 的 hidden_size_for/dtype_for 类方法，集中形状决策逻辑。

- python/sglang/srt/speculative/eagle_worker.py（模块 推测解码；类别 source；类型 core-logic）：替换 _draft_preprocess_idle 和 forward_draft_extend_after_decode 中的硬编码形状为类方法调用。
- python/sglang/srt/speculative/eagle_draft_extend_cuda_graph_runner.py（模块 推测解码；类别 source；类型 core-logic）：CUDA 图 extend buffer 初始化中删除原有的 EAGLE-3 aux 分支判断，统一使用 EagleDraftExtendInput.hidden_size_for/dtype_for。
- python/sglang/srt/speculative/eagle_draft_cuda_graph_runner.py（模块 推测解码；类别 source；类型 core-logic）：CUDA 图 decode buffer 初始化中替换硬编码的 model_runner.model_config.spec_hidden_size 为类方法。

关键符号：_draft_runner_of, EagleDraftInput.hidden_size_for, EagleDraftInput.dtype_for, EagleDraftExtendInput.hidden_size_for, EagleDraftExtendInput.dtype_for

关键源码片段

python/sglang/srt/speculative/eagle_info.py

核心文件：新增 _draft_runner_of 辅助函数以及 EagleDraftInput 和 EagleDraftExtendInput 的 hidden_size_for/dtype_for 类方法，集中形状决策逻辑。

```
def _draft_runner_of(worker):
    """Draft model_runner accessor that handles v1/v2 worker naming."""
    # v1（EAGLEWorker 等）将 draft model_runner 暴露为 model_runner；
    # v2（EagleDraftWorker 等）暴露为 draft_runner。
    return (
        worker.draft_runner if hasattr(worker, "draft_runner") else worker.model_runner
    )


class EagleDraftInput:
    # ... 其他字段和方法，重点关注以下类方法

    @classmethod
    def hidden_size_for(cls, worker) -> int:
        """Decode-phase `hidden_states` width: draft self-chain output
        (draft model writes its own last hidden back via `capture_for_decode`
        and the draft loop)."""
        # 返回 draft 模型输出的 hidden states 大小（spec_hidden_size）
        return _draft_runner_of(worker).model_config.spec_hidden_size

    @classmethod
    def dtype_for(cls, worker) -> torch.dtype:
        return _draft_runner_of(worker).model_config.dtype
```

评论区精华

无 review 评论。作者在 PR body 详细说明了三种 spec 算法类别（Chain / Non-chain / No-consume）及本 PR 覆盖范围，并规划了后续 PR 的演进路径。

- 后续路线图规划 (other): 本 PR 被接受合并，作为第一步。

风险与影响

- 风险：本 PR 仅为重构，不改变任何运行时行为，回归风险较低。但需注意：若存在自定义 worker 子类覆盖了相关模型配置属性，统一调用类方法后可能得到与之前不同的形状（概率极低）。测试覆盖方面，没有新增专用测试，但 CI 上 speculative decoding 测试均通过。
- 影响：对用户无影响。对开发维护者：hidden_states 形状计算逻辑集中化，降低了后续修改的成本和出错概率，为未来功能（如 STANDALONE 模式跳过 hidden_states 分配）铺平道路。
- 风险标记：缺少测试覆盖，重构可能暴露子类适配问题

关联脉络

- 暂无明显关联 PR